

Optimization of Problems with Multiple Objectives using The Multi-Verse Optimization Algorithm

S. Mirjalili ¹, P. Jangir ², S. Z. Mirjalili ³, S. Saremi ¹, I. N. Trivedi³

¹ *School of Information and Communication Technology, Griffith University, Australia*

² *Lukhdhirji Engineering College, Morbi-Rajkot, Gujarat, India*

³ *School of Electrical Engineering and Computing, University of Newcastle, Callaghan, NSW 2308, Australia*

Abstract

This work proposes the multi-objective version of the recently proposed Multi-Verse Optimizer (MVO) called Multi-Objective Multi-Verse Optimizer (MOMVO). The same concepts of MVO are used for converging towards the best solutions in a multi-objective search space. For maintaining and improving the coverage of Pareto optimal solutions obtained, however, an archive with an updating mechanism is employed. To test the performance of MOMVO, 80 case studies are employed including 49 unconstrained multi-objective test functions, 10 constrained multi-objective test functions, and 21 engineering design multi-objective problems. The results are compared quantitatively and qualitatively with other algorithms using a variety of performance indicators, which show the merits of this new MOMVO algorithm in solving a wide range of problems with different characteristics.

Keywords: Multi-objective Optimization; Multi-verse Optimizer; Constrained multi-objective optimization; MVO; MOMVO

1. Introduction:

Engineering optimization deals with finding optimal solutions for a given engineering problem. In the past, designing an engineering system would require a large number of tedious experiments. This means that a designer would have to build prototype to be able to test different designs and find the optimal one. For instance, to find an optimal shape for a car, a designer would have to build multiple prototypes of the car with changing the body and measuring the wind resistance in a wind tunnel. Obviously, such process would have been very expensive and required massive amount of human involvement in both design and test phases. Due to the natural errors that humans make, the design process was very unreliable as well.

After the invention of computers, the field of computer-aided design came to the existence and allowed engineers to solve engineering problems with computers. In the early steps, designers would employ computer to simulate the problem and all the factors around it. Therefore, there was little need for building prototype and testing in real environments. For instance, a computer program was able to construct the 3D shape of a car's body and simulating the drag of wind. In this case, the designers concentrated mostly on the design part rather than simulation. This significantly reduced the cost of the whole design process and amount of human involvement. However, such computer-aided engineering design was semi-automated because the computer only simulated the system and its surrounding environment.

Another significant progress in the field of engineering optimization and computer-aided design was to utilize computer to both design and simulate the system. This approach, which is still of the best techniques in the field of engineering optimization, requires the least human involvements. A designer needs to only formulate, prepare, and set up the problem for computers. In addition, an automatic problem solving techniques should be employed to find the optimal design. This is where optimization algorithms come in. Optimization algorithms command the computer how to find optimal solutions for optimization problems. This main

advantages of this approach are the lower error, less human involvement, and cheaper design cost. However, it requires designing an optimization algorithm to address all the difficulties that are available in engineering design problems.

It is undeniable that the advantages of the last approach outweigh its drawbacks. The majority of the first optimization algorithms employed to solve engineering problems used gradient descent [1]. This means that they used to calculate the derivation of the problem to find an optimal solution starting from an initial design. After a while, designers realized that there are two main problems with gradient-based methods. For one, most of the real problems are simulated in computers with sophisticated mathematical equations with unknown or expensive derivation. For another, gradient descend may lead to a local solution and highly depends on the initial solution.

The process of finding a local optimum and assuming it as the global optimum by an algorithm is called local optima stagnation [2]. Due to the unknown shape of the search space of real-world engineering problems, local optima stagnation prevents gradient-based algorithms from being beneficial. This resulted in the proposal of stochastic optimization algorithms in the past. Such approaches, equipped with random components, promote slight or major changes during the optimization process to avoid local solutions. The main advantage of stochastic algorithms is the ability to avoid local solutions. However, stochastic behaviours make the results of such algorithms different in each runs. It should be noted here that stochastic algorithms mostly do not need gradient information and consider a system as a black box.

Stochastic optimization algorithms are divided into two main groups: individual-oriented versus population-oriented. In the former class, a single design for a given engineering design problem is created and improved during optimization. By contrast, the latter method solves a problem with a collection of solutions. The nature of population-oriented algorithms makes them more reliable due to the higher probability of local optima avoidance [3]. However, they need more

function evaluation and require special mechanisms to exchange information between solutions. The most popular individual-oriented algorithms are Simulated Annealing (SA) [4] and hill climbing [5], while Genetic Algorithms (GA) [6], Ant Colony Optimization (ACO) [7, 8], and Particle Swarm Optimization (PSO) [9, 10] are the most well-regarded population-oriented algorithms.

Despite the flexibility and the ease of stochastic optimization algorithms, optimization of real engineering design problems involves addressing several and often conflicting difficulties: constraints, uncertainties, expensive objection function, dynamic objective function, multiple objectives, etc. This paper concentrates on handling multiple objectives [11], which is one of the most important characteristics of engineering design problems. Optimization of such problems is challenging due to the existence of multiple criteria to compare the solutions. The field that concentrates on handling multiple objectives using stochastic optimization techniques is Evolutionary Multi-Objective Optimization (EMOO) [12]. This field is one of the most popular fields in the recent years and its algorithms are being used widely in both science and industry [13-15].

The most well-regarded algorithms in EMOO are Non-Dominated Sorting Genetic Algorithm (NSGA) [16], Multi-Objective Particle Swarm Optimization (MOPSO) [17], and Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [18]. These algorithms are all population-oriented and designed based on Pareto optimality concepts. Pareto optimal dominance allows such techniques to compare solutions considering multiple objectives. Since there is more than one “optimal solution” for a multi-objective problem, multi-objective algorithms mostly utilize a storage to store best solutions (Pareto optimal solutions) obtained so far and improve this set during optimization.

Most of the recent algorithms have been equipped with suitable mechanisms to solve problems with multiple objectives [3, 19-21]. However, it has been logically proved by the well-known No-

Free-Lunch (NFL) theorem [22] that *none* of these algorithms is able to solve *all* optimization problems. This means that there is room for improving the current algorithms or proposing new ones to better solve a certain group of problems. The current algorithms are mostly suitable for unconstrained problems and unable to address different types of constraints without special components. This work targets constrained problems and proposes the multi-objective version of the recently proposed Multi-Verse Optimization (MVO) [23] algorithm to solve both unconstrained and constrained problems. The MVO has been inspired from one of the theories on the existence of multiple universes and mimics the interaction between them. The Multi-Objective MVO (MOMVO) is proposed that uses an archive and leader selection mechanisms in a similar manner to those in the MOPSO algorithm. The rest of the paper is organized as follows:

Section 2 presents the related works and reviews the literature. Section 3 briefly introduces the single-objective version of the MVO algorithm. Also, the multi-objective MVO algorithms are proposed in Section 3. Section 4 includes the results, discussions, and analysis of the results. Eventually, Section 5 concludes the paper and presents a couple of research directions for future.

2. Related works and literature review

This section first starts with the definitions of multi-objective optimization, Pareto optimality, Pareto dominance, Pareto optimal set, and Pareto optimal front. Different multi-objective optimization techniques are then discussed and reviewed.

2.1. Multi-objective optimization

As the name implies, multi-objective optimization addresses multiple objectives. It can be written as a minimization problem as follows:

$$\text{Minimize :} \quad F(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_o(\vec{x})\} \quad (2.1)$$

$$\text{Subject to :} \quad g_i(\vec{x}) \geq 0, \quad i = 1, 2, \dots, m \quad (2.2)$$

$$h_i(\vec{x}) = 0, \quad i = 1, 2, \dots, p \quad (2.3)$$

$$L_i \leq x_i \leq U_i, \quad i = 1, 2, \dots, n \quad (2.4)$$

where n is the number of variables, o is the number of objective functions, m is the number of inequality constraints, p is the number of equality constraints, g_i is the i -th inequality constraints, h_i indicates the i -th equality constraints, and $[L_i, U_i]$ are the boundaries of i -th variable.

It should be noted that there is only one solution for a single-objective problem due to the existence of unary objective. Also, comparison of solutions can be easily made by relational operators: $<$ or \leq for comparing algorithms in a minimization problem and $>$ or \geq in a maximization problem. In a multi-objective problem, however, there is no longer one best solution, which is because of considering multiple objectives. In fact, a set of solutions that contains the best trade-offs between objectives is the answer for a multi-objective problem.

Relational operators are ineffective in comparing solutions with respect to multiple objectives. The most common operator in the literature is Pareto optimal dominance, which is defined as follows for minimization problems [24]:

$$\forall i \in \{1, 2, \dots, k\}: f_i(\vec{x}) \leq f_i(\vec{y}) \quad \wedge \quad \exists i \in \{1, 2, \dots, k\}: f_i(\vec{x}) < f_i(\vec{y}) \quad (2.5)$$

where $\vec{x} = (x_1, x_2, \dots, x_k)$ and $\vec{y} = (y_1, y_2, \dots, y_k)$.

For maximization problems, Pareto optimal dominance is defined as follows:

$$\forall i \in \{1, 2, \dots, k\}: f_i(\vec{x}) \geq f_i(\vec{y}) \quad \wedge \quad \exists i \in \{1, 2, \dots, k\}: f_i(\vec{x}) > f_i(\vec{y}) \quad (2.5)$$

where $\vec{x} = (x_1, x_2, \dots, x_k)$ and $\vec{y} = (y_1, y_2, \dots, y_k)$.

This equation shows that a solution is better than another in a multi-objective search space if it is equal in all objective and better in at least one of the objectives (see Fig. 1). Pareto optimal

dominance is denoted with $<$ and $>$. With these two operators, solutions can be easily compared and differentiated.

Pareto optimality refers to the solutions that are the best when using Pareto dominance. They are also referred as non-dominated solutions. There is a set of best non-dominated solutions for every multi-objective problem called true Pareto optimal solution set. This set is mathematically defined as follows:

$$PS := \{\vec{x}, \vec{y} \in S \mid \nexists \vec{y} < \vec{x}\} \quad (2.6)$$

where S indicates a set of solutions and PS shows the Pareto solution set.

Eq. (2.6) shows that the Pareto optimal solution set is a set in which no solution is dominated by another. In other words, all solutions in this set are non-dominated. An example of Pareto optimal set is illustrated in Fig. 1.

The main objective of a multi-objective optimization algorithm is to find this set. Another common term and set in multi-objective optimization is Pareto optimal front. This set includes the projection of Pareto optimal solutions in the objective space. This is illustrated in Fig. 1 for a minimization problem. This figure also shows a dominated and a non-dominated solution in both parameter and objective spaces.

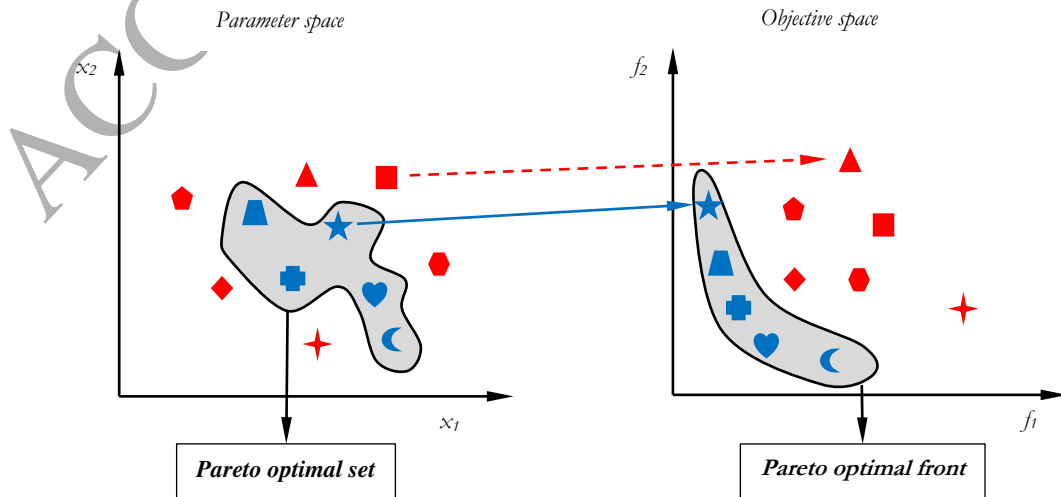


Figure 1. Parameter space and objective space in multi-objective optimization. Both objectives should be minimized in this example.

2.2. Related works

Multi-objective optimization utilizing stochastic population-oriented algorithm was first proposed by Schaffer in 1984. The main idea was to compare solutions using Pareto optimal dominance operators and approximate the Pareto optimal set for a given problem instead of using relational operators and determining only one solution. After the proposal of this idea, the field of EMOO was born and attracted much attention. Due to the similarity of multi-objective and single-objective population-oriented algorithms, they inherit all the advantages of the techniques mentioned in the preceding section. The most popular multi-objective population-oriented algorithms in the literature are: MOEA/D, NSGA-II, MOPSO, Pareto Archived Evolutionary [25], and Prato-frontier Differential Evolution (PDE) [26]. Such technique can be divided to three main classes: aggregation (*a priori*) methods [27], *a posteriori* methods [28], and interactive methods [29].

In the first class, aggregation-based approached, multiple objectives are aggregated to a single objective utilizing a set of weights, which define the importance of each objective from a decision maker point of view. After the aggregation phase, a single-objective algorithm is employed to find the optimal solution. If the decision maker's preference is not available, Pareto optimal set can be constructed with changing the weights and re-running the single-objective algorithm. The main drawback of this class of algorithms is that an evenly distributed set of weights does not result in an even set of Pareto optimal solutions [30]. Due to the summation of objectives using positive weights, in addition, such methods are not able to find non-convex regions in the Pareto optimal front. However, solving multi-objective problems with this approach is fairly straightforward and does not require algorithm modifications. It is worth

mentioning here that there are several improvements in the literature of aggregation methods [31], yet the above-mentioned disadvantages are not alleviated completely.

On the other hand, *a posteriori* technique maintains the multi-objective formulation of a multi-objective problem and approximates the Pareto optimal solutions. Therefore, there is no need for additional weights and aggregation. Such algorithms determine the Pareto optimal solutions set in just one run, but decision making is done after the optimization. This emphasizes the importance of solutions' distribution across all objectives to give decision makers several options. Needless to say, maintaining the multi-objective formulation requires algorithms modification and addressing multiple objectives, which are often in conflict. The algorithms in this class are very popular in the literature.

The last class of multi-objective stochastic optimization algorithm includes interactive methods. This group is somehow between the preceding two groups, in which decision making preferences are considered and incorporated during optimization. An interactive method keeps the multi-objective formulation but pauses the execution of the program and fetch the decision makers' preferences. This assists algorithm not to explore undesirable regions of the search space and Pareto optimal front. However, it needs human involvement and make it naturally slower than the aggregation and *a posteriori* algorithms.

It is worth mentioning here that there are two main goals when determining the Pareto optimal front using *a posteriori* algorithms: convergence and coverage. The former refers to the speed and accuracy of an algorithm in approximating Pareto optimal solutions. Obviously, the ultimate objective here is to find a very accurate Pareto optimal solution set with the least possible number of function evaluation. The latter concept, however, refers to the distribution of Pareto optimal solutions across all objectives. In this case, an effective algorithm is the one that finds a uniformly distributed Pareto optimal front to give decision makers freedom of decision making from a lot of diverse designs. The main challenge here is that convergence and coverage are in

conflict, so an algorithm should be able to efficiently balance them to be useful in solving real-world multi-objective problems.

In addition, another important aspect of multi-objective optimization algorithms is the execution time to determine an accurate estimation of true Pareto optimal solutions. Parallel multi-objective optimization algorithms are the best solutions to decrease the run time [20, 32].

In this work, the main concentration is on the proposal of a *a posteriori* method based on the recently proposed MVO algorithm. This algorithm uses an archive, which is very similar to that in MOPSO and PAES. Also, a mechanism is employed to choose non-dominated solutions to improve the coverage. The next section first presents the MVO algorithm. Then, the new multi-objective algorithm is proposed.

3. Multi-objective Multi-Verse Optimization algorithm (MOMVO):

3.1. Multi-Verse Optimizer

The MVO algorithm [23] mimics one of the theories in physics on the existence of multiple universe in the world. The main inspiration is the interaction of multiple universes via white hole, black hole, and worm hole. In this theory, objects are transferred from a universe through a tunnel from a white hole to a black hole. Also, worm holes are able to move objects from one corner of a universe to another without a need for a white or black hole.

MVO is a population-based algorithms and can be considered in the family of evolutionary algorithms. The optimization process starts with set of candidate solutions. Each candidate solution is analogous to a universe and variables are considered as objects in the universe. Similarly to other evolutionary algorithms, MVO has operators to combine the solutions and keep the best one(s). In order to combine the solutions, white and black holes are randomly created in the universes and causes movement of objects. It should be noted that each universe is evaluated with an objective function and its objective value is considered as the inflation rate.

The inflation rate is the growing speed of a universe and calculated proportional to the objective function.

What guarantees the improvement of solutions over the course of generation is the higher probability of white holes proportional to the inflation rate. The better inflation rate, the higher probability that a universe has to own white holes. By contrast, the existence of black holes is inversely proportional to the inflation rate. These rules cause the flow of variables from better universes to worse ones. The main difference between the Genetic Algorithm (GA) and MVO is that GA perform cross over on parts of parents and create a child with only two of them, while MVO allows any solution to contribute to the creation of new solutions. The while-black hole tunnels cause exploration of search space due to changing solutions suddenly.

Another main difference between GA and MVO is that MVO has elitism and save the best solution obtained so far. However, the GA algorithm discards all solutions created in the previous generation(s). It is worth mentioning here that the worm holes create tunnels between the best solution obtained so far and any solutions in the population. Therefore, the mutation of MVO is not 100% random as that of GA. In other words, the best solution obtained so far has the potential to contribute in the creation of any new solutions. This mechanism improves the exploitation of this MVO algorithm.

The main equation in the MVO algorithm is as follows [23]:

$$x_i^j = \begin{cases} \begin{cases} x_j + TDR + (ub_j - lb_j \times r_4 + lb_j) & r_3 < 0.5 \\ x_j + TDR + (ub_j - lb_j \times r_4 + lb_j) & r_3 \geq 0.5 \end{cases} & r_2 < WEP \\ x_i^j & r_2 \geq WEP \end{cases} \quad (3.1)$$

where x_i^j indicates the j -th variable in the i -th solution, x_j shows the j -th variable in the best solution, WEP is the worm hole existence probability, TDR indicates the travelling distance rate, lb_j is the lower bound of j -th dimension, ub_j is the upper bound of j -th dimension, $r_2 - r_4$ are

random values. Note that distribution of r_2-r_4 can be tuned to emphasize exploration or convergence.

There are several parameters in this equation, but the most important ones are WER and TDR. These two variables change adaptively to balance exploration and exploitation. WER stands for Wormhole Existence Rate and TDR stands for Travelling Distance Rate. The WER should be increased to emphasise exploitation proportional to the number of generation. By contrast, the TDR is better to be decreased to reduce the magnitude of changes when using the worm holes.

In the original paper of MVO, it has been proven that this algorithm is able to solve optimization problems effectively and provides very competitive results compared to the current algorithms. The main advantage of this algorithm compared to the existing ones is the high exploration. Several mechanisms abruptly and randomly change the solutions, which results in a very high exploration and consequently local optima avoidance. This motivated our attempts to propose the multi-objective version of this algorithm in the following subsection.

3.2. Multi-Objective Multi-Verse Optimizer (MOMVO)

In order to develop the multi-objective version of MVO, an archive is first integrated to this algorithm. This archive is very similar to those in MOPSO [33] and PAES. It is a repository to store the best non-dominated solutions obtained so far. The search mechanism in MOMVO is very similar to that is MVO, in which solutions are improved using white, black, and worm holes. Due to the existence of multiple best solutions, however, the white holes and specially worm holes should be chosen from the archive.

In order to select solutions from the archive to establish tunnels between solutions, we employ a leader selection mechanism. In this approach, the crowding distance between each solution in the archive is first selected and the number of solutions in the neighbourhood is counted as the measure of coverage or diversity. MOMVO is required to select solutions with a roulette wheel

from the less populated regions of the archive using the following equation to improve the distribution of solutions in the archive across all objectives:

$$P_i = c/N_i \quad (3.2)$$

where c is a constant and should be greater than 1 and N_i is the number of solutions in the vicinity of the i -th solution. This parameter c should be greater than one and remain constant since it decreases the fitness of the hypercubes containing more particles and it may be considered as a way of fitness sharing [34].

This equation gives a high probability to solutions in the less populated areas of the areas to contribute in the improvement of others. This automatically gravitates solutions towards the less populated regions of the archive and eventually improves the coverage of solutions in the Pareto optimal front obtained.

Obviously, the archive is able to accommodate a limited number of non-dominated solutions, and it might become full during the optimization process. So, there should be a mechanism to remove undesirable solutions from the archive. An undesired solution is the one with many neighbouring solutions, so we require the archive to throw away such solutions. The following equation, which is the inverse of Eq. (2.2), is employed to give high probability to undesired solutions in the archive to be discarded by the MOMVO algorithm.

$$P'_i = N_i/c \quad (3.3)$$

where c is a constant and should be greater than 1 and N_i is the number of solutions in the vicinity of the i -th solution. Note that the value of c needs to remain constant when calculating the probability for all the segments. This parameter should be greater than one since it increases the fitness of the hypercubes containing more particles.

With the above operators the MOMVO algorithm is able to store Pareto optimal solutions in the archive and improve them over the course of iterations. This algorithm complies with the following principles when comparing and adding a solutions to the archive in a manner similar to that in [33]:

- If a new solution dominates any solutions in the archive, it should be replaced by them immediately.
- If a new solution does not dominate a solution in the archive, it should be discarded and not allowed to enter the archive.
- If a solution is non-dominated with respect to all solutions in the archive, it should be added to the archive.
- If the archive is full, an undesired solution should be deleted and the new non-dominated solution should enter the archive.

To compare all the solutions in MOMVO, the concepts of Pareto optimality and Pareto optimal solution are used. This is because in a multi-objective search space, solutions cannot be compared with relational operators ($<$, \leq , $>$, \geq , $=$, \neq). Therefore, we need to use Pareto dominance to find out if a solution is better than another in a multi-objective search space or not. The Pareto dominance is able to consider more than one comparison criteria and allows us to find Pareto optimal solutions.

The main difference between MOMVO algorithm and NSGA-II is the archive. In NSGA-II, solutions are sorted and chosen based on their domination level. Although this mechanism promotes exploration, it might damage good non-dominated solutions. The main motivation of using an archive for MOMVO was to reduce the probability of damaging non-dominated solutions during optimization. The archive maintains the best non-dominated solutions. Since we required the solutions in MOMVO to updates their variables abruptly using several operators,

this algorithm theoretically provides a superior exploration and convergence compared to the NSGA-II algorithm.

The MOMVO algorithm is able to be potentially more promising than other archive-based techniques such as SPEA2 due to different mechanism for abruptly changing the solutions during optimization. SPEA2 use cross over and mutation in a pool of solutions in very iteration. This biases the search more towards the archive members compared to the MOMVO algorithm. In the proposed algorithm, exchanging variables between solutions might occur between a solution and an archive member or two non-dominated solutions in the search space. This increases the exploration, which might negatively impact the convergence as a potential drawback for MOMVO. We tried to give 50% chance of selecting an archive member or a non-dominated solution in the search space to balance exploration and exploitation.

The MOMVO algorithm starts approximating the true Pareto optimal front for a given multi-objective optimization problem with a random set of solutions. Each solution is assigned with multiple objective values. At first, all the non-dominated solutions are chosen and inserted to the archive. After the first iteration, the algorithm repeatedly updates the position of solutions using Eq. (3.1). This equation gives 50% change to exchange variables with an archive member or one of the non-dominated solutions in the current population. The former mechanism promotes the exploitation of the best Pareto optimal solutions obtained so far. The latter operator, however, improves the exploration of the search space. The process of improving the solutions is continued until the satisfaction of a termination condition. The coverage of solutions across all objectives are improved by selecting solutions from the less populated regions of the archive as well.

In the following section, a number of test functions and real engineering design problems are employed to benchmark the performance of the proposed MOMVO algorithms.

4. Results on test functions

4.1. Experimental set up

In order to benchmark the performance of the proposed algorithm a variety of test functions are used. They can be divided to 3 main groups as follows:

- Unconstrained multi-objective test functions (ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, DEB1, DEB2, UF1-UF10, LAU, LIS, MUR, QUAG, REND1, REND2, BINH1, SCH2, OKA1, LZ01-LZ09, SCH1, FON1, FON2, KUR, VNT1, VNT2, VNT3, and POL) [35-39] [40-45]
- Constrained multi-objective test functions (CONSTR, TNK, SRN, BNH, OSY, KITA, BEL, BNH4, VNT4, and CF1) [46-48] [49, 50]
- Engineering design multi-objective problems (gear train, pressure vessel, helical compression spring, two-bar truss, welded beam, four-bar truss, disk brake, speed reducer, CNC machine tool, tool spindle, I-beam, cantilever, multiple disk clutch, brushless DC wheel motor, safety isolating transformer, vibrating platform, satellite heat pipe, three-bar truss, bulk carrier, car crash, and metal cutting tool. [51-61]

Note that some of the test functions have linear fronts, so they can be solved by aggregating the objectives since the objectives are not in conflict. However, there are even standard test functions in the literature with linear front (*e.g.* UF5, UF6, and U7 in the CEC test suite) because such test functions are very helpful for benchmarking the coverage of an algorithm since there is no intrinsic bias towards different regions.

In order to compare the algorithms on the above case studies, we employ both quantitative and qualitative metrics. For quantifying the convergence, we have chosen Generational Distance (GD) proposed by Veldhuizen in 1998 [62] and Inverted Generational Distance (IGD) [63]. In order to quantify the distribution of Pareto optimal solutions obtained (coverage), Spacing (SP) metric proposed by Schott [64] [65] and Maximum Spread (M) proposed by Zitzler [66] have been employed. The mathematical equations of these performance indicators are as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^{no} d_i^2}}{no} \quad (4.1)$$

where no is the number of obtained Pareto optimal solutions and d_i indicates the Euclidean distance between the i -th Pareto optimal solution obtained and the closest true Pareto optimal solution in the reference set. Note that the Euclidean distance is calculated in the objective space.

$$IGD = \frac{\sqrt{\sum_{i=1}^{nt} (d'_i)^2}}{nt} \quad (4.2)$$

where nt is the number of true Pareto optimal solutions and d'_i indicates the Euclidean distance between the i -th true Pareto optimal solution and the closest Pareto optimal solution obtained in the reference set.

$$SP \triangleq \sqrt{\frac{1}{no-1} \sum_{i=1}^{no} (\bar{d} - d_i)^2} \quad (4.3)$$

where \bar{d} is the average of all d_i , no is the number of Pareto optimal solutions obtained, and $d_i = \min_j (|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$ for all $i, j=1, 2, \dots, n$.

The lower the value for this metric, the better the coverage is. The second coverage measure is called:

$$M = \sqrt{\sum_{i=1}^o \max(d(a_i, b_i))} \quad (4.4)$$

where o is the number of objectives, and $d()$ calculates the Euclidean distance, a_i is the maximum value in the i -th objective, and b is the minimum value in the i -th objective

To see the results qualitatively, the best Pareto optimal fronts obtained by the MOMVO algorithm on the case studies are illustrated in the appendix A. Each algorithm was run 30 times and the results are presented and discussed in the following section. For results verification, MOMVO is compared to a wide range of algorithms including MOEA/D, NSGA-II, and SPEA-II in the next subsection.

Table 1. Results of the multi-objective algorithms (using GD) on the unconstrained on test functions employed

Algorithm	ZDT1 (2 objs)		ZDT2 (2 objs)		ZDT3 (2 objs)		ZDT4 (2 objs)		ZDT6 (2 objs)	
	Ave	Std.	Ave	Std.	Ave	Std.	Ave	Std.	Ave	Std.
MOMVO	0.00173	0.00005	0.000948	0.00005	0.005076	0.00009	0.002463	0.00006	0.000652	0.00002
MOEA/D-DE	0.00413	0.00008	0.004422	0.00006	0.005129	0.00011	0.003971	0.00009	0.003322	0.00024
NSGA-II-DE	0.005186	0.00028	0.004133	0.00020	0.006200	0.00041	0.077546	0.09002	0.003273	0.00027
Algorithm	DTLZ1 (2 objs)		DTLZ2 (2 objs)		DTLZ3 (2 objs)		DTLZ4 (2 objs)		DTLZ5 (2 objs)	
	Ave	Var.	Ave	Var.	Ave	Var.	Ave	Var.	Ave	Var.
MOMVO	1.1E-06	1.6E-11	0.00026	4.8E-13	0.0017	2.2E-03	0.0019	4.32E-05	0.0019	4.32E-05
MOEA/D-DE	4.2E-05	1.8E-10	0.00035	2.8E-11	0.1079	2.2E-02	0.0258	6.66E-04	0.0258	6.66E-04
NSGA-II	0.12580	2.2E-01	0.00029	1.3E-09	0.8250	3.6E-01	0.0435	2.53E-04	0.0435	2.53E-04
Algorithm	DTLZ6 (2 objs)		DTLZ7 (2 objs)		DTLZ8 (2 objs)		DTLZ9 (2 objs)		DEB1 (2 objs)	
	Ave	Var.	Ave	Var.	Ave	Var.	Ave	Var.	Ave	Var.
MOMVO	0.0034	0.0018	0.0020	0.00011	0.00798	5.34E-08	3.88E-5	4.79E-10	3.88E-5	4.79E-10
MOEA/D-DE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
NSGA-II	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Algorithm	DTLZ10 (3 objs)		DTLZ11 (3 objs)		DTLZ12 (3 objs)		DTLZ13 (3 objs)		DTLZ14 (3 objs)	
	Ave	Var.	Ave	Var.	Ave	Var.	Ave	Var.	Ave	Var.
MOMVO	0.00102	2.1E-12	0.00056	4.6E-12	0.0010	4.4E-05	0.0185	2.14E-04	0.0185	2.14E-04
MOEA/D-DE	0.00111	3.1E-11	0.00096	8.7E-11	0.0048	3.5E-04	0.0457	1.06E-03	0.0457	1.06E-03
NSGA-II	0.00447	1.2E-04	0.00162	3.7E-08	1.5488	5.06E+0	0.0326	2.83E-04	0.0326	2.83E-04
Algorithm	DTLZ15 (3 objs)		DTLZ16 (3 objs)		DTLZ17 (3 objs)		DTLZ18 (3 objs)		DEB2 (2 objs)	
	Ave	Std.	Ave	Std.	Ave	Std.	Ave	Std.	Ave	Std.
MOMVO	0.00014	2.3E-05	9.0E-05	2.7E-06	1.8E-03	8.17E-05	1.35E-04	2.38E-05	1.35E-04	2.38E-05
SPEA-II	0.00018	8.4E-12	0.28130	N/A	0.01180	N/A	N/A	N/A	N/A	N/A
NSGA-II	0.00017	1.1E-09	0.09410	N/A	0.01110	N/A	N/A	N/A	N/A	N/A
Algorithm	UF1 (2 objs)		UF2 (2 objs)		UF3 (2 objs)		UF4 (2 objs)		UF5 (2 objs)	
	Ave	Std.	Ave	Std.	Ave	Std.	Ave	Std.	Ave	Std.
MOMVO	0.010576	0.00572	0.010020	0.00300	0.02169	0.03557	0.040464	0.00248	0.279502	0.19521
MOEA/D-DE	0.067801	0.00703	0.034738	0.00555	0.02453	0.02798	0.060091	0.00819	0.564877	0.22891
NSGA-II-DE	0.011549	0.00548	0.010039	0.00127	0.06703	0.03799	0.037238	0.00146	0.315271	0.21944
Algorithm	UF6 (2 objs)		UF7 (2 objs)		UF8 (2 objs)		UF9 (2 objs)		UF10 (2 objs)	
	Ave	Std.	Ave	Std.	Ave	Std.	Ave	Std.	Ave	Std.
MOMVO	0.049160	0.05561	0.007524	0.00167	0.635041	0.11475	1.319607	0.54863	1.72438	0.40508
MOEA/D-DE	0.306746	0.06107	0.043162	0.00624	0.429362	0.08932	0.397282	0.04172	0.39666	0.10734

NSGA-II-DE	0.052228	0.09148	0.006895	0.00133	2.858333	0.28673	1.666083	0.42798	6.43363	0.54881
Algorithm	LAU (2 objs)		LIS (2 objs)		MUR (2 objs)		QUAG (2 objs)			
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.059	0.001	0.0005	0.0001	0.00019	0.00001	0.0009	0.00058		
IGMOEA-5	0.109	0.012	0.002	0.000	0.000	0.000	0.003	0.000		
NSGA-II	0.089	0.002	0.002	0.000	0.000	0.000	0.001	0.000		
Algorithm	REND1 (2 objs)		REND2 (2 objs)		BINH1 (2 objs)		SCH2 (2 objs)			
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.0007	0.00001	0.0001	0.00017	0.002	0.0001	0.00158	0.00314		
IGMOEA-5	0.001	0.000	0.001	0.000	0.011	0.001	N/A	N/A		
NSGA-II	0.001	0.000	0.001	0.000	0.007	0.000	0.3096	0.02175		

Table 2. Results of the multi-objective algorithms (using IGD) on the unconstrained test functions employed

Algorithm	ZDT1 (2 objs)		ZDT2 (2 objs)		ZDT3 (2 objs)		ZDT4 (2 objs)		ZDT6 (2 objs)	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.000504	0.00006	0.000509	0.00005	0.009057	0.00006	0.000413	0.00004	0.000405	0.00004
MOEA/D-DE	0.004136	0.00009	0.004100	0.00007	0.009168	0.00007	0.003965	0.00009	0.003167	0.00024
NSGA-II-DE	0.004501	0.00028	0.004533	0.00026	0.006070	0.00037	0.077979	0.08876	0.013749	0.00021
Algorithm	DTLZ1 (2 objs)		DTLZ2 (2 objs)		DTLZ3 (2 objs)		DTLZ4 (2 objs)			
	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>
MOMVO	6.6E-05	5.9E-14	1.5E-04	2.2E-16	0.0001	3.57E-05	0.00018	2.50E-05		
MOEA/D-DE	1.3E-04	1.0E-12	1.7E-04	1.3E-15	0.0311	2.10E-03	0.03373	1.57E-04		
NSGA-II	4.0E-04	1.6E-07	1.8E-04	6.7E-11	0.2036	2.40E-02	0.02000	1.41E-04		
Algorithm	DTLZ5 (2 objs)		DTLZ6 (2 objs)		DTLZ7 (2 objs)		DEB3 (2 objs)			
	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>
MOMVO	0.0004	0.0001	0.0007	0.00012	0.0000054	8.32E-12	8.58E-04	1.59E-08		
MOEA/D-DE	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A		
NSGA-II	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A		
Algorithm	DTLZ1 (3 objs)		DTLZ2 (3 objs)		DTLZ3 (3 objs)		DTLZ4 (3 objs)			
	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>	<i>Ave</i>	<i>Var.</i>
MOMVO	4.2E-04	1.5E-12	7.5E-04	3.4E-12	0.0015	2.4E-06	1.9E-04	3.77E-06		
MOEA/D-DE	7.6E-04	1.5E-11	8.0E-04	7.6E-12	0.0017	7.1E-06	9.7E-03	4.07E-06		
NSGA-II	6.7E-04	4.5E-08	7.9E-04	9.5E-10	0.0528	1.0E-03	5.0E-03	6.0E-06		
Algorithm	DTLZ5 (3 objs)		DTLZ6 (3 objs)		DTLZ7 (3 objs)		OKA1 (2 objs)			
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	1.1E-04	1.1E-05	1.6E-04	1.7E-05	1.1E-03	1.8E-04	9.66E-03	1.29E-05		
SPEA-II	4.3E-03	3.8E-04	6.4E-01	5.9E-02	6.3E-02	2.8E-03	N/A	N/A		
NSGA-II	5.5E-03	3.7E-04	6.5E-01	6.1E-02	7.7E-02	4.1E-03	N/A	N/A		
Algorithm	LZ01 (2 objs)			LZ02 (2 objs)			LZ03 (2 objs)			
	<i>Ave</i>	<i>Std.</i>	<i>Best</i>	<i>Ave</i>	<i>Std.</i>	<i>Best</i>	<i>Ave</i>	<i>Std.</i>	<i>Best</i>	
MOMVO	0.0014	0.0000	0.0014	0.0010	7.8E-5	0.0009	0.0010	0.0002	0.0006	
MOEA/D-DE	0.0015	0.0000	0.0015	0.0028	0.0004	0.0023	0.0068	0.0099	0.0022	
NSGA-II-DE	0.0044	0.0000	0.0044	0.0349	0.0066	0.0203	0.0296	0.0030	0.0228	
Algorithm	LZ04 (2 objs)			LZ05 (2 objs)			LZ06 (2 objs)			
	<i>Ave</i>	<i>Std.</i>	<i>Best</i>	<i>Ave</i>	<i>Std.</i>	<i>Best</i>	<i>Ave</i>	<i>Std.</i>	<i>Best</i>	
MOMVO	0.0010	0.0001	0.0008	0.0009	0.0001	0.0006	0.0066	0.0012	0.0054	
MOEA/D-DE	0.0040	0.0014	0.0025	0.0127	0.0069	0.0073	0.0289	0.0014	0.0276	
NSGA-II-DE	0.0288	0.0021	0.0251	0.0288	0.0031	0.0244	0.0680	0.0072	0.0522	
Algorithm	LZ07 (2 objs)			LZ09 (3 objs)						
	<i>Ave</i>	<i>Std.</i>	<i>Best</i>	<i>Ave</i>	<i>Std.</i>	<i>Best</i>				
MOMVO	0.0034	0.0027	0.0007	0.0009	0.0001	0.0008				
MOEA/D-DE	0.0049	0.0063	0.0015	0.0035	0.0008	0.0025				
Algorithm	UF1 (2 objs)		UF2 (2 objs)		UF3 (2 objs)		UF4 (2 objs)		UF5 (2 objs)	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.009550	0.00170	0.009040	0.00016	0.003773	0.00276	0.001433	0.00018	0.033936	0.00314
MOEA/D-DE	0.064926	0.00164	0.029768	0.00031	0.029823	0.03411	0.058549	0.00675	0.113243	0.03029
NSGA-II-DE	0.042377	0.01228	0.015293	0.00203	0.247844	0.08995	0.035261	0.00089	0.034925	0.00328
Algorithm	UF6 (2 objs)		UF7 (2 objs)		UF8 (2 objs)		UF9 (2 objs)		UF10 (2 objs)	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.006534	0.0022	0.007690	0.00117	0.024627	0.00033	0.031639	0.00954	0.369872	0.08259
MOEA/D-DE	0.267101	0.0812	0.041148	0.00039	0.122289	0.00103	0.193243	0.03263	0.366985	0.03651
NSGA-II-DE	0.104944	0.04595	0.022305	0.00330	1.036505	0.29149	0.070979	0.01610	0.361609	0.11337

Algorithm	SCH1 (2 objs)		FON1 (2 objs)		FON2 (2 objs)		KUR(2 objs)	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	3.48E-3	4.3E-05	9.84E-4	4.6E-05	1.64E-3	1.50E-04	8.59E-3	1.5E-04
SPEA-II	1.66E-2	1.1E-04	4.66E-3	7.1E-05	4.18E-3	3.2E-04	3.41E-2	7.1E-04
NSGA-II	1.87E-2	4.3E-04	5.56E-3	1.7E-04	5.15E-3	4.8E-04	4.23E-2	1.9E-03

Algorithm	VNT1 (3 objs)		VNT3 (2 objs)		VNT3 (3 objs)		POL (2 objs)	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	1.81E-2	1.19E-4	2.12E-3	3.63E-5	8.28E-3	1.60E-4	6.32E-4	2.42E-4
SPEA-II	1.27E-1	2.7E-03	1.23E-2	3.7E-04	3.24E-2	1.0E-03	5.31E-2	1.1E-03
NSGA-II	1.58E-1	6.7E-03	2.31E-2	2.3E-03	4.98E-2	3.2E-03	6.96E-2	5.7E-03

4.2. Results on unconstrained and constrained test functions

The results of algorithms on unconstrained multi-objective test functions are presented in Table 1 and Table 2. As per the results of these two tables, it is evident that MOMVO is able to outperform other algorithms on the majority of unconstrained test functions. IGD and GD performance indicators quantify the convergence of multi-objective algorithms. Therefore, the results show that MOMVO benefits from improved convergence. The best Pareto optimal fronts obtained by the algorithm is shown in the appendix A. It can be seen that that Pareto optimal solutions estimated are of high distribution across all objectives.

The results of algorithms on constrained multi-objective test functions are shown in Table 3. To solve these kinds of problems, we use a death penalty function as the main constraint handling technique for MOMVO. The results show that the MOMVO outperforms other algorithms on the majority of case studies as well. The performance indicators quantitatively prove this, and the Pareto fronts in the appendix A show that this algorithm is better qualitatively as well. The distribution of the solutions is very high, showing that the MOMVO is able to handle constraints and find Pareto optimal solutions in different feasible regions.

4.3. Solving real-worlds problems using MOMVO

In order to confirm the efficiency and effectiveness of the proposed optimization algorithm on real-world problems, this subsection solves a variety of engineering design problems using the

proposed MOMVO. The results of algorithm on multi-objective engineering design problems are provided in Table 4 and 5. These cases studies are the most challenging ones employed in this work and able to effectively benchmark the performance of MOMVO. Inspecting the results in Table 4 and 5 and best Pareto optimal fronts obtained in the appendix A, it is evident that MOMVO is able to find very accurate approximation of the true Pareto optimal solutions for real problems as well. The high coverage can be observed in the figure.

Table 3. Results of the multi-objective algorithms on constrained benchmark test problems

CONSTR test problem (2 objs)								
Algorithm	GD		Metric of spread		Metric of spacing		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	8.1386E-04	5.3142E-05	0.39873	5.0074E-02	0.0287	0.0013	1.3E-04	1.14E-05
MOPSO	4.5437E-03	6.8558E-04	0.94312	3.6719E-01	N/A	N/A	N/A	N/A
NSGA-II	5.1349E-03	2.4753E-04	0.54863	2.7171E-02	0.0437	0.0041	N/A	N/A
TNK test problem(2 objs)								
Algorithm	GD		Metric of spread		Metric of spacing		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	3.1765E-04	4.5897E-05	0.67482	2.4030E-02	0.003	0.0001	4.9E-04	5.69E-05
MOPSO	5.0877E-03	4.5564E-04	0.79363	5.1029E-02	N/A	N/A	N/A	N/A
NSGA-II	4.0488E-03	4.3465E-04	0.82286	2.8678E-04	N/A	N/A	N/A	N/A
SRN test problem (2 objs)								
Algorithm	GD		Metric of spread		Metric of spacing		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	4.2709E-05	2.5599E-06	0.3749	2.4572E-02	0.6273	0.118	5.25E-04	1.58E-06
MOPSO	2.7623E-03	2.0794E-04	0.6655	7.2196E-02	N/A	N/A	N/A	N/A
NSGA-II	3.7069E-03	5.1034E-04	0.3869	2.5115E-02	1.586	0.133	N/A	N/A
BNH test problem (2 objs)								
Algorithm	GD		Metric of spread		Metric of spacing		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	3.6839E-04	4.5175E-05	0.3512	2.2963E-02	0.3789	0.032	8.9E-04	2.21E-05
MOPSO	N/A	N/A	N/A	N/A	0.6941	0.038	N/A	N/A
NSGA-II	N/A	N/A	N/A	N/A	0.7756	0.072	N/A	N/A
OSY test problem (2 objs)								
Algorithm	GD		Metric of spread		Metric of spacing		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	5.56E-02	4.89E-02	0.3945	2.9596E-02	0.5125	0.081	3.4E-02	1.27E-02
MOPSO	9.68E-02	7.18E-02	N/A	N/A	0.522	0.095	N/A	N/A
NSGA-II	9.89E-01	9.78E-01	N/A	N/A	1.14	0.275	N/A	N/A
KITA test problem (2 objs)								
Algorithm	GD		Metric of spread		Metric of spacing		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.0035	3.52E-02	0.00691	0.0089	0.1352	0.1179	3.6E-04	1.785E-05
MOPSO	0.0467	0.0535	0.09925	0.1176	0.3184	0.4778	N/A	N/A
NSGA-II	0.04	0.044	0.7863	0.1951	0.1462	0.1518	N/A	N/A
BEL test problem								
BNH4 test problem (2 objs)			VNT4 test problem (2 objs)			CF1 test problem (2 objs)		
Algorithm	IGD (2 objs)		IGD		IGD		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	3.60E-3	5.3E-05	9.75E-2	1.1E-03	3.78E-2	1.2E-04	2.70E-3	5.78E-04
SPEA-II	3.74E-2	3.0E-04	5.87E-1	1.6E-02	1.11E-1	2.0E-03	N/A	N/A

NSGA-II	4.61E-2	1.6E-03	7.09E-1	3.5E-02	1.77E-1	1.2E-02	N/A	N/A
---------	---------	---------	---------	---------	---------	---------	-----	-----

Table 4. Results of the multi-objective algorithms on engineering design problems

Problems	Gear train design problem (2 objs)				Pressure vessel design problem (2 objs)			
Algorithm	Minimum F1		Minimum F2		Minimum F1		Minimum F2	
	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>
MOMVO	2.358E-09	39	3.32E-02	21	42.020	8171.0	2.779E+05	6.099E+07
MOWCA	4.50E-09	43	7.32E-01	12	N/A	N/A	N/A	N/A
NSGA-II	1.8E-08	37	5.01E-01	13	38.982	7329.34	3.223 E+05	6.366 E+07
Helical compression spring design problem (2 objs)	GD				IGD			
Algorithm	<i>Max.</i>	<i>Min.</i>	<i>Median</i>	<i>Std.</i>	<i>Max.</i>	<i>Min.</i>	<i>Median</i>	<i>Std.</i>
MOMVO	8.87E-04	4.97 E-04	7.02 E-04	7.8 E-06	13.69 E-04	6.96 E-05	8.62 E-04	2.13 E-05
MOWCA	9.16 E-04	5.27 E-04	7.18 E-04	9.8 E-05	15.70 E-04	7.58 E-04	9.33 E-04	1.50 E-04
Two-bar truss design problem (2 objs)	GD				IGD			
Algorithm	<i>Max.</i>	<i>Min.</i>	<i>Median</i>	<i>Std.</i>	<i>Max.</i>	<i>Min.</i>	<i>Median</i>	<i>Std.</i>
MOMVO	0.000485	0.000243	0.000401	0.0000645	0.000368	0.000183	0.000221	0.0000592
E-PSO	0.000408	0.000328	0.000381	0.000021	0.000428	0.000277	0.000312	0.000036
NSGA-II	0.000338	0.000216	0.000277	0.000028	0.000266	0.000222	0.000245	0.000011
MOPSO-CD	0.001220	0.000276	0.000376	0.000179	0.001020	0.000493	0.000621	0.000110
Welded beam design problem (2 objs)	GD		Metric of spread		Metric of spacing		IGD	
Algorithm	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.01587	0.00809	0.20587	0.00349	0.0388	0.0079	0.00426	0.00287
MOPSO	0.04909	0.02821	0.22478	0.09280	N/A	N/A	N/A	N/A
NSGA-II	0.16875	0.08030	0.88987	0.11976	N/A	N/A	N/A	N/A
Four-bar truss design problem (2 objs)	GD		Metric of spread		Metric of spacing		IGD	
Algorithm	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.2491	0.0156	0.174	0.0088	1.1017	0.158	0.0281	0.1165
MOPSO	0.3741	0.0422	N/A	N/A	2.5303	0.227	N/A	N/A
NSGA-II	0.3601	0.0470	N/A	N/A	2.3635	0.255	N/A	N/A
Disk brake design problem (2 objs)	GD		Metric of spread		Metric of spacing		IGD	
Algorithm	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.0132	0.02681	0.42058	0.04671	0.2283	0.0541	0.00565	0.00098
MOPSO	0.0244	0.12314	0.46041	0.10961	N/A	N/A	N/A	N/A
NSGA-II	3.0771	0.10782	0.79717	0.06608	N/A	N/A	N/A	N/A
Speed reducer design problem (2 objs)	GD		Metric of spread		Metric of spacing		IGD	
Algorithm	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.85792	0.0098	0.9767	0.0626	1.8765	1.739	0.43067	0.04121
MOPSO	0.98831	0.1789	N/A	N/A	16.685	2.696	N/A	N/A
NSGA-II	9.8437	7.0810	N/A	N/A	2.7654	3.534	N/A	N/A
CNC Machine tool design problem (2 objs)	Minimum F1		Minimum F2		GD		IGD	
Algorithm	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>

MOMVO	0.853	9.487	1.125	2.112	0.0038	0.0042	0.00611	0.00275
NSGA-II	0.854	9.488	1.125	2.113	N/A	N/A	N/A	N/A
Tool spindle design problem (2 objs)								
Algorithm	Minimum F1		Minimum F2		GD		IGD	
	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>	<i>FS</i>	<i>Ave</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	474465.86	0.037149	1646075.69	0.016612	1296549.78	0.2033	0.3271	0.02549
PSO-EO	474653.67	0.037186	1646089.55	0.016613	1171435.88	N/A	N/A	N/A
MOTS	497644.10	0.037839	1485169.00	0.016894	987524.900	N/A	N/A	N/A
I- beam design problem (2 objs)								
Algorithm	Minimum F1		Minimum F2		GD		IGD	
	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>	<i>FS</i>	<i>Ave</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	127.65	0.05507	848.6	0.005907	726.54	0.1470	0.1334	0.01646
PSO-EO	127.71	0.06424	850.00	0.005909	722.29	N/A	N/A	N/A
MOTS	143.52	0.03700	678.21	0.006640	534.69	N/A	N/A	N/A

Table 5. Results of the multi-objective algorithms on engineering design problems

Cantilever beam design problem (2 objs)								
Algorithm	Minimum F1		Minimum F2		GD		IGD	
	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.4415	0.002021	3.056	4.21E-05	0.00038	0.0001	0.00052	0.00018
MOPSO	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
NSGA-II	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Multiple disk clutch brake design problem (2 objs)								
Algorithm	Minimum F1		Minimum F2		GD		IGD	
	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.3137	11.6500	2.0521	3.2579	0.1234	0.0123	0.04199	0.00875
NSGA-II	0.4704	11.7617	2.0948	3.3505	N/A	N/A	N/A	N/A
Brushless DC Wheel Motor design problem (2 objs)								
Algorithm	GD		Metric of spread		Metric of spacing		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	3.71E-3	4.27E-3	0.8211	0.07845	0.05513	0.0047	2.42E-3	1.57E-3
SPEA-II	8.78E-2	4.22E-2	N/A	N/A	N/A	N/A	N/A	N/A
NSGA-II	8.51E-2	1.70E-2	N/A	N/A	N/A	N/A	N/A	N/A
Safety Isolating Transformer design problem (2 objs)								
Algorithm	GD		Metric of spread		Metric of spacing		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	1.43952E-3	1.58196E-4	0.9612	0.05938	0.0678	0.0059	3.17E-3	3.30E-4
MOSQP	6.54731E-2	9.85432E-3	N/A	N/A	N/A	N/A	N/A	N/A
NSGA-II	2.65719E-2	7.67234E-3	N/A	N/A	N/A	N/A	N/A	N/A
Vibrating platform design problem (2 objs)								
Algorithm	Pareto optimal point		Metric of spacing		GD		IGD	
	<i>F1</i>		<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	80/3052		0.1872	0.0605	0.1470	0.0426	0.05516	0.0038
EO	60/3052		N/A	N/A	N/A	N/A	N/A	N/A
Satellite Heat Pipe design problem (2 objs)								
Algorithm	Minimum F1		Maximum F2		GD		IGD	
	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	25.87	0.322	27.52	0.3820	0.0321	0.0338	0.044823	0.01125
BA	26.70	N/A	N/A	0.3818	N/A	N/A	N/A	N/A
GA	26.71	N/A	N/A	0.3812	N/A	N/A	N/A	N/A
HS	26.70	N/A	N/A	0.3810	N/A	N/A	N/A	N/A
Three-bar truss design problems (2 objs)								
Algorithm	Minimum F1		Minimum F2		GD		IGD	
	<i>F1</i>	<i>F2</i>	<i>F1</i>	<i>F2</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	2.6685E+2	6.3460E-7	19.13E+2	0.8228E-7	0.0423	0.0012	0.02545	0.003987
MONLP	4.5288E+2	4.2962E-7	6.507E+2	2.7141E-7	N/A	N/A	N/A	N/A
Bulk Carrier Design problems (3objs)								
Algorithm	HV		Metric of spread		GD		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>

MOMVO	0.444852	0.000524	0.13584	0.0652	0.000103	0.000057	0.00908	0.005213
SAMO	0.444701	0.001275	N/A	N/A	N/A	N/A	N/A	N/A
NSGA-II	0.275908	0.034077	N/A	N/A	0.000106	0.000014	0.01153	0.001270
Car Crash design problems (3objs)								
Algorithm	HV		Metric of spread		GD		IGD	
	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>	<i>Ave</i>	<i>Std.</i>
MOMVO	0.010570	0.000015	0.2782	0.3065	0.2351	0.10251	0.01608	0.004737
SAMO	0.010410	0.000077	N/A	N/A	N/A	N/A	N/A	N/A
NSGA-II	0.006365	0.001425	N/A	N/A	N/A	N/A	N/A	N/A
Metal Cutting tool design problems (3 objs)								
Algorithm	Minimum F1			Minimum F2			Minimum F3	
	<i>F1</i>	<i>F2</i>	<i>F3</i>	<i>F1</i>	<i>F2</i>	<i>F3</i>	<i>F1</i>	<i>F2</i>
MOMVO	0.1627	0.1194	3.951	0.176	0.073	3.243	2.737	1.067
NSGA-II	0.163	0.125	2.161	0.181	0.073	1.098	2.731	1.065

In the previous paragraphs, the convergence of algorithms was compared quantitatively, whereas the coverage was reported and compared qualitatively. To quantify the coverage, we employ Execution Time (ET), Metric of Spread (MS), and Inverse Hyper Volume (IHV) metrics to show the efficiency and effectiveness of proposed MOMVO algorithm compared to NSGA-II and MODE [67-70] on some of the real world problems [51-61] employed above. The results are given in Table 6 (test functions) and Table 7 (engineering design problems). Note that N/A in the tables stands for Not Applicable, which refers to the algorithms that we did not find any results for a particular engineering design problem in the literature.

Table 6. Results of different metrics, Hyper Volume (HV) and Execution Time (ET), for MOMVO, NSGA-II and MOPSO algorithms.

Problems	HV in m^3 (avg.)			ET in second (avg.)		
	MOMVO	NSGA-II	MOPSO	MOMVO	NSGA-II	MOPSO
ZDT1 (2 objs)	0.999854	0.994704	0.820794	3.02587	8.8836	97.1898
ZDT2 (2 objs)	0.998732	0.995543	0.602551	3.10279	11.4737	76.6658
ZDT3 (2 objs)	0.998579	0.992548	0.627398	5.87542	9.3890	80.0386
ZDT4 (2 objs)	0.993287	0.989682	0.633132	6.28831	15.2137	125.8195
ZDT6 (2 objs)	0.998678	0.872076	0.994543	7.69872	9.5596	35.9135
DTLZ1 (3 objs)	0.999158	0.992282	0.995180	7.63112	9.8537	25.6417
DTLZ2 (3 objs)	0.999872	0.995327	0.972694	5.87924	7.6157	242.6490
DTLZ3 (3 objs)	0.987558	0.985328	0.977411	8.35789	10.9689	21.7793
DTLZ4 (3 objs)	0.999475	0.975525	0.993028	7.65328	8.0930	145.7864
DTLZ5 (3 objs)	0.996879	0.987040	0.994401	7.93873	8.0209	186.3189
DTLZ6 (3 objs)	0.999548	0.974175	0.980990	6.83278	9.5265	255.5049
DTLZ7 (3 objs)	0.983258	0.963346	0.931623	7.93571	8.5004	185.6130

Table 7. Results of different metrics Metric of Spread (MS), Inverse Hyper Volume (IHV), and Execution Time (ET) with MOMVO, NSGA-II and MODE algorithms for engineering design problems.

Problems	MS in m (avg. \pm std.)			IHV in m^{-3} (avg. \pm std.)			ET in second (avg. \pm std.)		
	MOMVO	NSGA-II	MODE	MOMVO	NSGA-II	MODE	MOMVO	NSGA-II	MODE

Welded beam	0.20587±	0.88987±	0.58607±	6.89E-5±	1.3E-3±	5.39E-4±	0.32416±	0.46880±	0.37520±
design (2 objs)	0.003490	0.119760	0.043660	6.8354E-4	3.543E-3	1.016E-3	0.017600	0.024510	0.019400
Speed reducer	0.51489±	0.79717±	0.84041±	2.75E-4±	1.3E-3±	5.39E-4±	0.41807±	0.46260±	0.42500±
design (2 objs)	0.048590	0.066080	0.200850	2.357E-3	3.543E-3	1.016E-3	0.005689	0.008760	0.006710
Disk brake	0.51670±	0.66542±	0.61511±	9.89E-4±	2.1E-3±	1.41E-3±	0.11482±	0.16880±	0.13780±
design (2 objs)	0.062600	0.014310	0.045810	3.854E-5	7.450E-4	2.689E-4	0.002486	0.006610	0.007160
Bulk Carrier	0.13584±	N/A	N/A	3.28E-4±	3.2E-3±	1.99E-3±	0.58732±	N/A	N/A
Design (3 objs)	0.065200			1.254E-4	1.590E-4	1.969E-4	0.087135		
Two-bar truss	0.7982±	0.93725±	0.87393±	3.58E-5±	N/A	N/A	0.25872±	0.30660±	0.26560±
design (2 objs)	0.024730	0.024250	0.050320	2.658E-4			0.022579	0.00876	0.026850

Table 6 and 7 indicate that the MOMVO algorithm outperforms other algorithms in all new metrics. This quantitative comparison shows that the MOMVO algorithm benefits from a good coverage as well. To avoid the variability of each algorithm, Wilcoxon signed-rank test method [71] and Friedman [72] are used to validate comparative study between optimization algorithms in Table 6 for the HV metric as an example. Wilcoxon signed-rank test conducted at 10% significance level ($p\text{-value} > 0.1$ rejects the null hypothesis) on the best results of algorithms with the population size of 60 and 30 independent runs. The results of this statistical test (average $p\text{-values}$ on all test cases based on HV metric) are shown in Table 8. Note that the following procedure is used to perform Wilcoxon signed-rank test:

- Step 1: Average values of the HV metric in Table 6 obtained by MOMVO and other algorithms (say A) are collected.
- Step 2: Sum of ranks for cases where MOMVO outperforms A and vice versa are used to compute R^+ and R^- .
- Step 3: To ascertain the significance of statistical test based on hypotheses, $p\text{-value}$ are computed. A Smaller value represents stronger evidence against the null hypothesis.

It is evident in the results of Table 8 that no significant differences were found between MOMVO and MOPSO in average, while MOMVO is clearly superior to NSGA-II. We observed that the superiority of MOMVO on three-objective problems tend to be more significant compared both MOPSO and NSGA-II. This shows that MOMVO is able to better handle difficulties in challenging multi-objective problems.

Table 8: Wilcoxon signed ranks test results based on HV metric from Table 6.

Comparison	R ⁺	R ⁻	Exact P-Value
MOMVO versus MOPSO	192	108	0.0942
MOMVO versus NSGA-II	198	102	0.0506

Another statistical test conducted to prove the significance of the results is Friedman aligned test. In this method, a value of location is computed based on the average of MS from Table 7 as the mean outturn obtained by all algorithms on each problem. Then, the variation between the outturn achieved by an algorithm and the value of locations is calculated. This step is repeated for each combination of algorithms and problems. The aligned observations are then ranked from 1 to k relative to each other. The results are presented in Table 9. Once more, it is apparent that MOMVO ranks first and outperforms both MODE and NSGA-II.

Table 9: Average ranking of algorithms by Friedman Aligned test based on MS obtained from Table 7.

Rank	Algorithm	Ranking
1	MOMVO	64.5762
2	MODE	79.3732
3	NSGA-II	86.3485

It is worth mentioning here that as a Pareto dominance-based algorithm, MOMVO becomes less effective proportional to the number of objectives. This is due to the fact that in problems with more than three objectives, a large number of solutions are non-dominated, so the archive becomes full quickly. Therefore, the MOMVO algorithm is suitable for solving problems with less than four objectives. Increasing the archive size might help to capture best non-dominated solutions in a many-objective problem, but there should be special operators and mechanisms to reliably solve such problems using the proposed MOMVO algorithm. The above results showed that MOMVO has the potential to show superior results in terms of both accuracy and execution time compared to other similar algorithms when solving the majority of three-objective case studies. This makes this algorithm potentially able to provide competitive results on many-objective problems. Solving such problems is out of the scope of this work, yet it is a valuable contribution for future works.

4.4. Comparison of MOMVO with recent algorithms

In the preceding section, the MOMVO algorithm was compared with well-known algorithms. To validate the performance of this algorithm even further, this section employs recently proposed algorithms. The algorithms are Multi objective Symbiotic organism search (MOSOS) [73] and Multi-objective colliding bodies optimization (MOCBO) [74]. The comparative results are given in Table 10 with GD and ET performance metrics. To prove the significance of the results, Quade [75] statistical test are performed for GD in Table 11. The results of these two tables show that the MOMVO provides very competitive results and tends to outperform MOSOS and MOCBO on the majority of test functions.

4.7. Discussion

In summary the results and discussions of this section show that the MOMVO algorithm provides very high convergence speed. This originates from the fact that the best solution always contributes to the improvement of other solutions significantly. The first component of Eq. (3.1) requires solutions to change their positions with respect to the best solution obtained so far. This mechanism is a bonus in most of problem since the solutions gravitate towards the best regions of the search space. This emphasizes exploitation of the best regions consequently. Since MOMVO uses the same search mechanisms of MVO, it inherits the high exploitation. In spite of choosing 'best solution' from the archive, the results show that this mechanism does not negatively impact the exploitation. It is worth mentioning here that high exploitation also results in high convergence. Convergence is accelerated in MOMVO proportional to the number of iterations since WEP in Eq. (3.1) is increased adaptively proportional the number of iterations. Improved exploitation and adaptively increasing convergence are of the main reasons why MOMVO reforms really well on simple unimodal multi-objective test problems.

Also, MOMVO benefits from high exploration. This originates from the main operators in MVO and MOMVO. Sudden changes in the solution, which come from the second component of Eq. (3.1), cause extensive exploration of the search space in MVO. Since MOMVO uses the same equation to update the solutions, it inherently shows improved exploration as well. Another reason of high explorative behaviour is due to the leader selection in MOMVO. The roulette wheel operator allows every solution to impact on other solutions. For one, this mechanism improves the exploration since the solutions face abrupt changes. For another, solutions are able to resolve local optima stagnation since poor solutions are also able to attract other solutions although with a very low probability.

Table 10: Comparison MOMVO, MOSOS and MOCBO algorithms for different unconstraint and constraint problems.

Problems	GD (avg.)			ET in second (avg.)		
	MOMVO	MOSOS	MOCBO	MOMVO	MOSOS	MOCBO
SCH1 (2 objs)	0.0018	0.0028	0.0031	4.2873	8.2135	5.4845
SCH2 (2 objs)	0.00158	0.0705	0.0932	4.5982	8.7015	5.9751
KUR (2 objs)	0.0051	0.0075	0.0083	5.2347	10.741	7.9531
FON2 (2 objs)	0.0017	0.0019	0.0022	7.8962	11.401	8.6606
ZDT1 (2 objs)	0.00173	0.3325	0.3337	3.025	8.235	3.143
ZDT2 (2 objs)	0.000948	0.0731	0.0729	3.102	8.234	3.1501
ZDT3 (2 objs)	0.005076	0.1022	0.0982	5.875	13.456	6.2846
ZDT4 (2 objs)	0.002463	0.5015	0.5078	6.288±	13.902	6.6922
SRN (2 objs)	4.2709E-05	0.0988	0.1018	8.3254	12.325	7.3251
CONSTR (2 objs)	8.1386E-04	0.5162	0.5202	5.1687	10.011	5.2252
TNK (2 objs)	3.1765E-04	0.1508±	0.1528	12.259	15.128	11.010
OSY (2 objs)	5.56E-02	0.1196	0.1210	11.919	20.212	12.210
BNH (2 objs)	3.6839E-04	0.1436	0.1498	7.5369	16.266	9.1544
KITA (2 objs)	0.0035	0.0368	0.0384	9.5234	14.382	10.324

Table 11: Average ranking of algorithms by Quade test results based on GD obtained Metric from Table 10.

Rank	Algorithm	Ranking
1	MOMVO	2.3465
2	MOSOS	2.5753
3	MOCBO	3.0956

Another observation was the high coverage of Pareto optimal solutions obtained by MOMVO on most of the test functions. This originates from the leader selection and archive maintenance mechanisms. MOMVO selects leaders from the least populated regions of the obtained Pareto

optimal front. This requires other solutions to explore and exploit the regions of the search space and consequently find non-dominated solutions in the less populated regions. In addition, non-dominated solutions are removed from the most populated regions when the archive is full. This also promotes balancing the distribution of non-dominated Pareto optimal solutions in the archive.

The performance of NSGA-II algorithm on the challenging engineering design problems was good as well. This is due to the non-dominated sorting and cross-over mechanisms in this algorithm, which results in high exploration of the search space. In real-world problems, there is a considerable number of local fronts that can be avoided by the NSGA-II algorithm. On the other hand, MOSOS and MOCBO did not perform really well on most of the test cases. This is due to the high exploitation of these algorithms. Both techniques belong to the family of swarm intelligence techniques, which intrinsically have a lower exploration compared to evolutionary algorithms. NSGA-II and MOMVO have been equipped with operators to require the solution to change abruptly which resulted in significantly better results.

5. Conclusion:

This work proposed the multi-objective version of the recently proposed MVO algorithm. Inspiring from the main mechanism of MOPSO, an archive and leader selection mechanism were integrated to MOMVO. There was also an archive maintenance to balance the distribution of the solutions in the archive when it is full. The contribution was the proposal of a new multi-objective algorithm capable of solving both unconstrained and constrained multi-objective problems. Due to the archive controller and leader selection mechanism developed, the proposed MOMVO algorithm was computationally cheaper than the current well-known algorithm as well.

After the proposal of MOMVO, it was tested on 80 case studies including unconstrained/constrained test functions and real engineering problems. The result showed that the proposed MOMVO algorithm is able to effectively approximate the true Pareto optimal fronts for all the case studies employed. All the fronts obtained were of high distribution as well. The analyses of the results showed that the superiority of the results originated from the improved exploration and exploitation in MOMVO.

As per the results and finding of the comprehensive comparative study of this work, we conclude that this algorithm has merits among the current multi-objective algorithms. MOMVO is suitable for problems with two and three objectives. Due to the very high exploration and local front avoidance of this algorithm, it can be readily applied to real-world problems with many local fronts. For future works, it is recommended to apply different constrained handling techniques to MOMVO. Also, the proposal of techniques to handle many objectives (more than four) using MOMVO is recommended.

References:

- [1] P. Ochs, R. Ranftl, T. Brox, and T. Pock, "Techniques for Gradient-Based Bilevel Optimization with Non-smooth Lower Level Problems," *Journal of Mathematical Imaging and Vision*, vol. 56, pp. 175-194, 2016.
- [2] K.-L. Du and M. Swamy, "Particle swarm optimization," in *Search and Optimization by Metaheuristics*, ed: Springer, 2016, pp. 153-173.
- [3] S. Chowdhury, J. Zhang, and A. Messac, "Avoiding premature convergence in a mixed-discrete particle swarm optimization (mdpso) algorithm," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1678.
- [4] L. Junghans and N. Darde, "Hybrid single objective genetic algorithm coupled with the simulated annealing optimization method for building optimization," *Energy and Buildings*, vol. 86, pp. 651-662, 2015.
- [5] L. Davis, "Bit-Climbing, Representational Bias, and Test Suite Design," in *ICGA*, 1991, pp. 18-23.
- [6] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Machine learning*, vol. 3, pp. 95-99, 1988.
- [7] A. Coloni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the first European conference on artificial life*, 1991, pp. 134-142.
- [8] T. Liao, K. Socha, M. A. M. de Oca, T. Stützle, and M. Dorigo, "Ant colony optimization for mixed-variable optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, pp. 503-518, 2014.

- [9] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, 1995, pp. 39-43.
- [10] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Mathematical Problems in Engineering*, vol. 2015, 2015.
- [11] K. Deb, K. Sindhya, and J. Hakanen, "Multi-objective optimization," in *Decision Sciences: Theory and Practice*, ed: CRC Press, 2016, pp. 145-184.
- [12] K. Deb, *Multi-objective optimization using evolutionary algorithms* vol. 16: John Wiley & Sons, 2001.
- [13] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, pp. 369-395, 2004.
- [14] M. G. C. Tapia and C. A. C. Coello, "Applications of multi-objective evolutionary algorithms in economics and finance: A survey," in *IEEE congress on evolutionary computation*, 2007, pp. 532-539.
- [15] C. A. C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," *Knowledge and Information systems*, vol. 1, pp. 269-308, 1999.
- [16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *International Conference on Parallel Problem Solving From Nature*, 2000, pp. 849-858.
- [17] C. C. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, 2002, pp. 1051-1056.
- [18] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, pp. 712-731, 2007.
- [19] E. De la Hoz, E. de la Hoz, A. Ortiz, J. Ortega, and A. Martínez-Álvarez, "Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps," *Knowledge-Based Systems*, vol. 71, pp. 322-338, 2014.
- [20] D. Kimovski, J. Ortega, A. Ortiz, and R. Baños, "Parallel alternatives for evolutionary multi-objective optimization in unsupervised feature selection," *Expert Systems with Applications*, vol. 42, pp. 4239-4252, 2015.
- [21] A. Martínez-Álvarez, S. Cuenca-Asensi, A. Ortiz, J. Calvo-Zaragoza, and L. A. V. Tejuelo, "Tuning compilations by multi-objective optimization: Application to apache web server," *Applied Soft Computing*, vol. 29, pp. 461-470, 2015.
- [22] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, pp. 67-82, 1997.
- [23] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural Computing and Applications*, vol. 27, pp. 495-513, 2016.
- [24] Y. Censor, "Pareto optimality in multiobjective problems," *Applied Mathematics and Optimization*, vol. 4, pp. 41-59, 1977.
- [25] J. Knowles and D. Corne, "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, 1999.
- [26] H. A. Abbass, R. Sarker, and C. Newton, "PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, 2001, pp. 971-978.
- [27] I. Y. Kim and O. De Weck, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation," *Structural and multidisciplinary optimization*, vol. 29, pp. 149-158, 2005.
- [28] J. Branke, T. Kaußler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization," *Advances in Engineering Software*, vol. 32, pp. 499-507, 2001.
- [29] W. S. Shin and A. Ravindran, "Interactive multiple objective optimization: Survey I—Continuous case," *Computers & Operations Research*, vol. 18, pp. 97-114, 1991.
- [30] T. W. Athan and P. Y. Papalambros, "A note on weighted criteria methods for compromise solutions in multi-objective optimization," *Engineering Optimization*, vol. 27, pp. 155-176, 1996.
- [31] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and multidisciplinary optimization*, vol. 41, pp. 853-862, 2010.
- [32] D. Kimovski, J. Ortega, A. Ortiz, and R. Baños, "Leveraging cooperation for parallel multi-objective feature selection in high- dimensional EEG data," *Concurrency and Computation: Practice and Experience*, vol. 27, pp. 5476-5499, 2015.

- [33] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on evolutionary computation*, vol. 8, pp. 256-279, 2004.
- [34] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proceedings of the 3rd international conference on genetic algorithms*, 1989, pp. 42-50.
- [35] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, pp. 173-195, 2000.
- [36] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable test problems for evolutionary multiobjective optimization*: Springer, 2005.
- [37] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report*, vol. 264, 2008.
- [38] L. Nguyen, L. T. Bui, and H. A. Abbass, "DMEA-II: the direction-based multi-objective evolutionary algorithm-II," *Soft Computing*, vol. 18, pp. 2119-2134, 2014.
- [39] G.-Q. Zeng, J. Chen, L.-M. Li, M.-R. Chen, L. Wu, Y.-X. Dai, *et al.*, "An improved multi-objective population-based extremal optimization algorithm with polynomial mutation," *Information Sciences*, vol. 330, pp. 49-73, 2016.
- [40] L. T. Bui, K. Deb, H. A. Abbass, and D. Essam, "Interleaving guidance in evolutionary multi-objective optimization," *Journal of Computer Science and Technology*, vol. 23, pp. 44-63, 2008.
- [41] L. T. Bui, J. Liu, A. Bender, M. Barlow, S. Wesolkowski, and H. A. Abbass, "Dmea: a direction-based multiobjective evolutionary algorithm," *Memetic Computing*, vol. 3, pp. 271-285, 2011.
- [42] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, pp. 284-302, 2009.
- [43] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multi-objective optimization," 2015.
- [44] A. Panda and S. Pani, "A Symbiotic Organisms Search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems," *Applied Soft Computing*, vol. 46, pp. 344-360, 2016.
- [45] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, pp. 761-776, 2015.
- [46] N. Srinivasan and K. Deb, "Multi-objective function optimisation using non-dominated sorting genetic algorithm," *Evolutionary Comp*, vol. 2, pp. 221-248, 1994.
- [47] T. T. Binh and U. Korn, "MOBES: A multiobjective evolution strategy for constrained optimization problems," in *The Third International Conference on Genetic Algorithms (Mendel 97)*, 1997, p. 27.
- [48] A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Structural optimization*, vol. 10, pp. 94-99, 1995.
- [49] M. Li, S. Yang, K. Li, and X. Liu, "Evolutionary algorithms with segment-based search for multiobjective optimization problems," *IEEE transactions on cybernetics*, vol. 44, pp. 1295-1313, 2014.
- [50] A. Sadollah, H. Eskandar, and J. H. Kim, "Water cycle algorithm for solving constrained multi-objective optimization problems," *Applied Soft Computing*, vol. 27, pp. 279-298, 2015.
- [51] C. C. Coello and G. T. Pulido, "Multiobjective structural optimization using a microgenetic algorithm," *Structural and Multidisciplinary Optimization*, vol. 30, pp. 388-403, 2005.
- [52] T. Ray and K. M. Liew, "A swarm metaphor for multiobjective design optimization," *Engineering optimization*, vol. 34, pp. 141-153, 2002.
- [53] F. Moussouni, S. Brisset, and P. Brochet, "Some results on the design of brushless DC wheel motor using SQP and GA," *International Journal of Applied Electromagnetics and Mechanics*, vol. 26, pp. 233-241, 2007.
- [54] K. Deb and R. Datta, "Hybrid evolutionary multi-objective optimization and analysis of machining operations," *Engineering Optimization*, vol. 44, pp. 685-706, 2012.
- [55] K. Deb and A. Srinivasan, "Monotonicity analysis, evolutionary multi-objective optimization, and discovery of design principles," *Indian Institute of Technology, Kanpur Genetic Algorithm Laboratory (KanGAL), Report*, 2006.

- [56] K. Deb and A. Srinivasan, "Innovization: Discovery of innovative design principles through multiobjective evolutionary optimization," in *Multiobjective Problem Solving from Nature*, ed: Springer, 2008, pp. 243-262.
- [57] M. J. Jeong, T. Kobayashi, and S. Yoshimura, "Extraction of design characteristics of multiobjective optimization—its application to design of artificial satellite heat pipe," in *International Conference on Evolutionary Multi-Criterion Optimization*, 2005, pp. 561-575.
- [58] M. Kotinis, "A particle swarm optimizer for constrained multi-objective engineering design problems," *Engineering Optimization*, vol. 42, pp. 907-926, 2010.
- [59] A. M. Nafchi and A. Moradi, "Constrained Multi-Objective Optimization Problems in Mechanical Engineering Design Using Bees Algorithm," *Journal of Solid Mechanics Vol.*, vol. 3, pp. 353-364, 2011.
- [60] T. Ray, K. TAI, and K. C. SEOW, "Multiobjective design optimization by an evolutionary algorithm," *Engineering Optimization*, vol. 33, pp. 399-424, 2001.
- [61] C.-L. Yu, Y. Lu, and J. Chu, "Multi-objective optimization with combination of particle swarm and extremal optimization for constrained engineering design," *WSEAS Trans Syst Control*, vol. 4, pp. 129-138, 2012.
- [62] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm research: A history and analysis," Citeseer1998.
- [63] M. R. Sierra and C. A. C. Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and ϵ -dominance," in *Evolutionary multi-criterion optimization*, 2005, pp. 505-519.
- [64] J. R. Schott, "Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization," DTIC Document1995.
- [65] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 8, pp. 256-279, 2004.
- [66] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications* vol. 63: Shaker Ithaca, 1999.
- [67] W. Gong, Z. Cai, and L. Zhu, "An efficient multiobjective differential evolution algorithm for engineering design," *Structural and Multidisciplinary Optimization*, vol. 38, pp. 137-157, 2009.
- [68] F. Qian, B. Xu, R. Qi, and H. Tianfield, "Self-adaptive differential evolution algorithm with α -constrained-domination principle for constrained multi-objective optimization," *Soft Computing*, vol. 16, pp. 1353-1372, 2012.
- [69] Y. Huo, Y. Zhuang, J. Gu, and S. Ni, "Elite-guided multi-objective artificial bee colony algorithm," *Applied Soft Computing*, vol. 32, pp. 199-210, 2015.
- [70] X. Dai, X. Yuan, and Z. Zhang, "A self-adaptive multi-objective harmony search algorithm based on harmony memory variance," *Applied Soft Computing*, vol. 35, pp. 541-557, 2015.
- [71] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, pp. 3-18, 2011.
- [72] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the american statistical association*, vol. 32, pp. 675-701, 1937.
- [73] D.-H. Tran, M.-Y. Cheng, and D. Prayogo, "A novel Multiple Objective Symbiotic Organisms Search (MOSOS) for time–cost–labor utilization tradeoff problem," *Knowledge-Based Systems*, vol. 94, pp. 132-145, 2/15/ 2016.
- [74] A. Panda and S. Pani, "Multi-objective Colliding Bodies Optimization," in *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, 2016, pp. 651-664.
- [75] D. Quade, "Using weighted rankings in the analysis of complete blocks with additive block effects," *Journal of the American Statistical Association*, vol. 74, pp. 680-683, 1979.

Appendix A: Best Pareto optimal fronts obtained by MOMVO

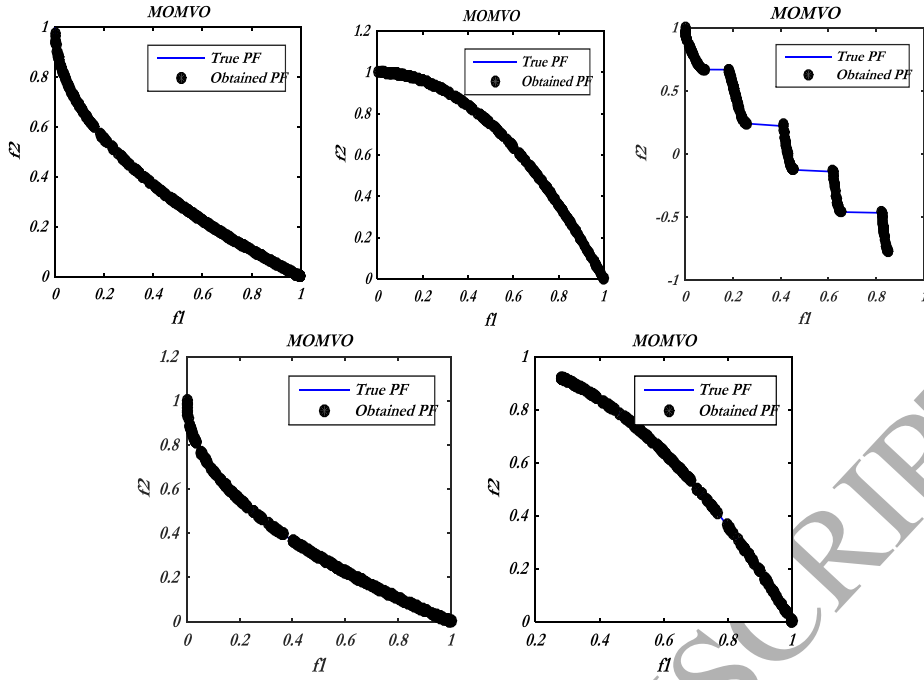


Figure A.1. Best Pareto optimal front obtained by the multi-objective MVO algorithm on ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 problems.

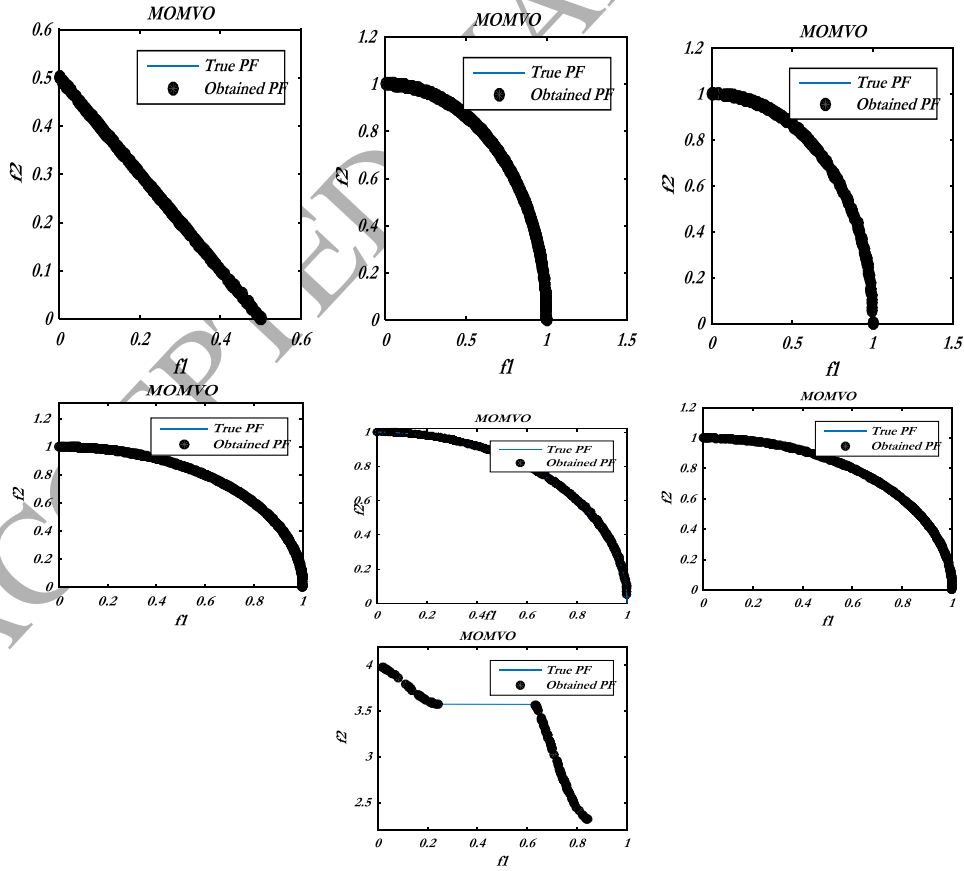


Figure A.2. Best Pareto optimal front obtained by the multi-objective MVO algorithm on 2-Dim. DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6 and DTLZ7 problems.

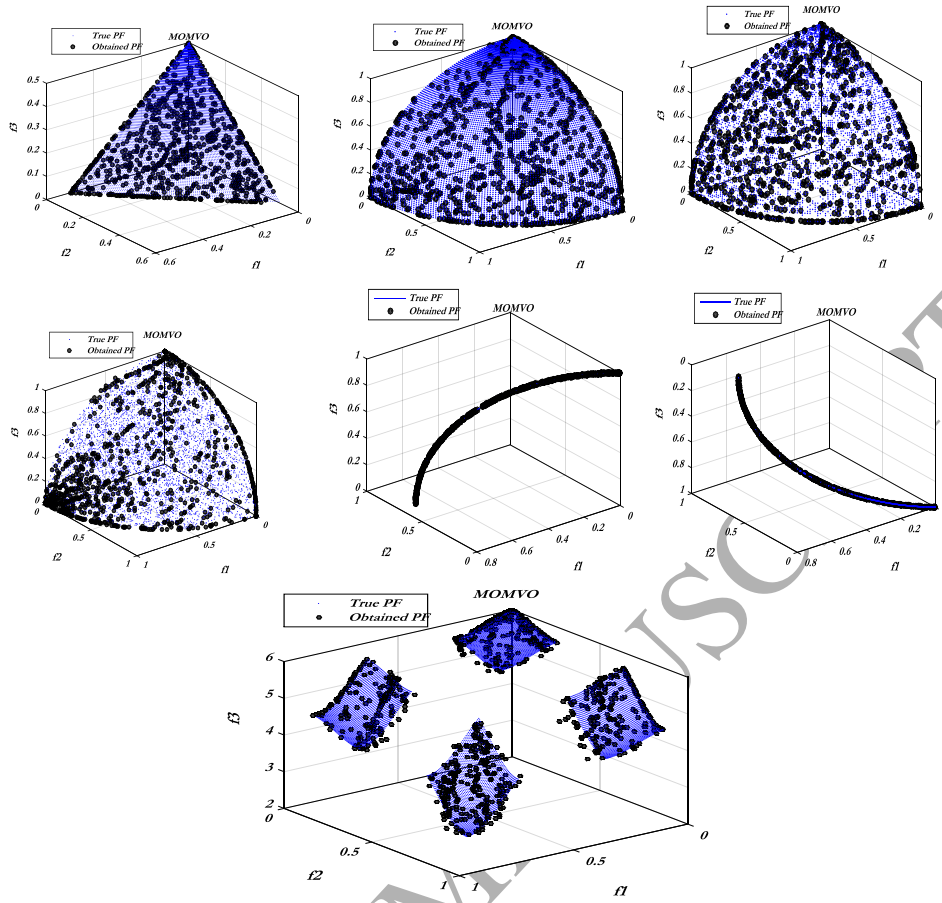


Figure A.3. Best Pareto optimal front obtained by the multi-objective MVO algorithm on 3-Dim. DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6 and DTLZ7 problems.

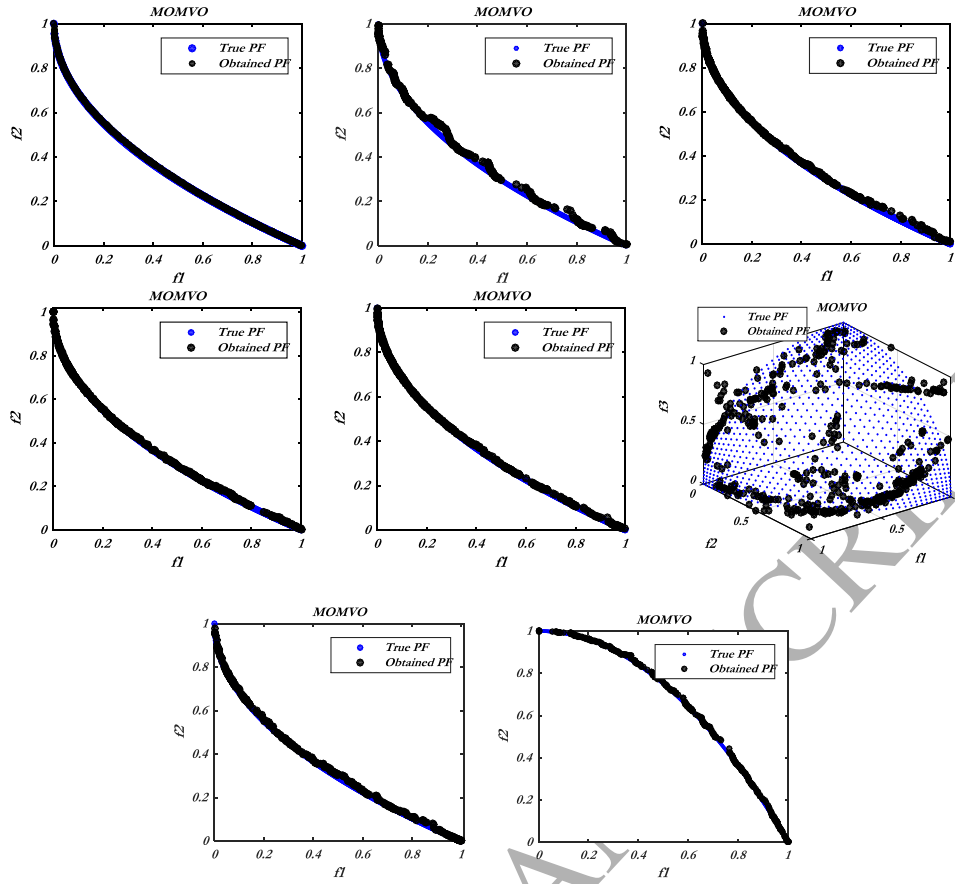


Figure A.4. Best Pareto optimal front obtained by the multi-objective MVO algorithm on LZ01, LZ02, LZ03, LZ04, LZ05, LZ06, LZ07 and LZ09 problems.

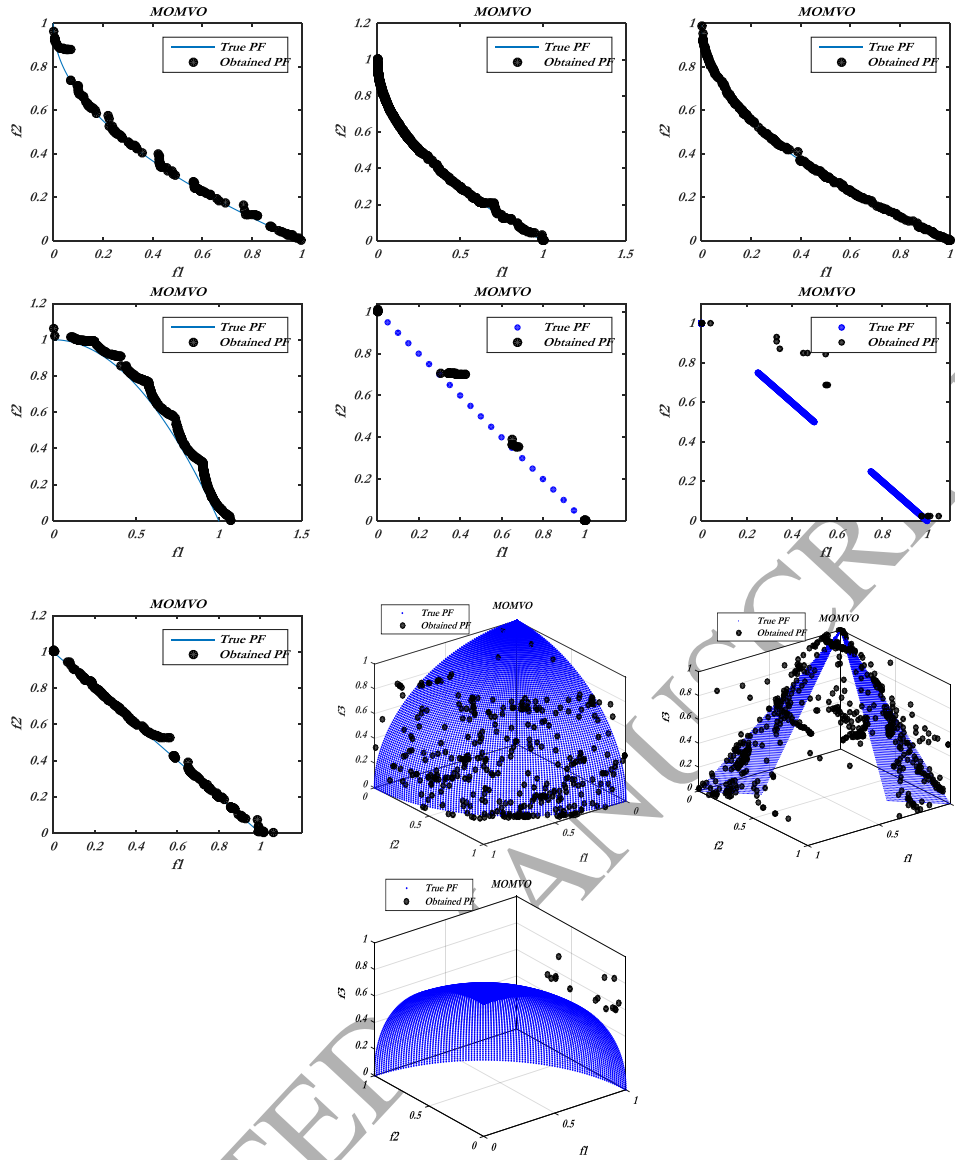


Figure A.5. Best Pareto optimal front obtained by the multi-objective MVO algorithm on UF1, UF2, UF3, UF4, UF5, UF6, UF7, UF8, UF9 and UF10 problems.

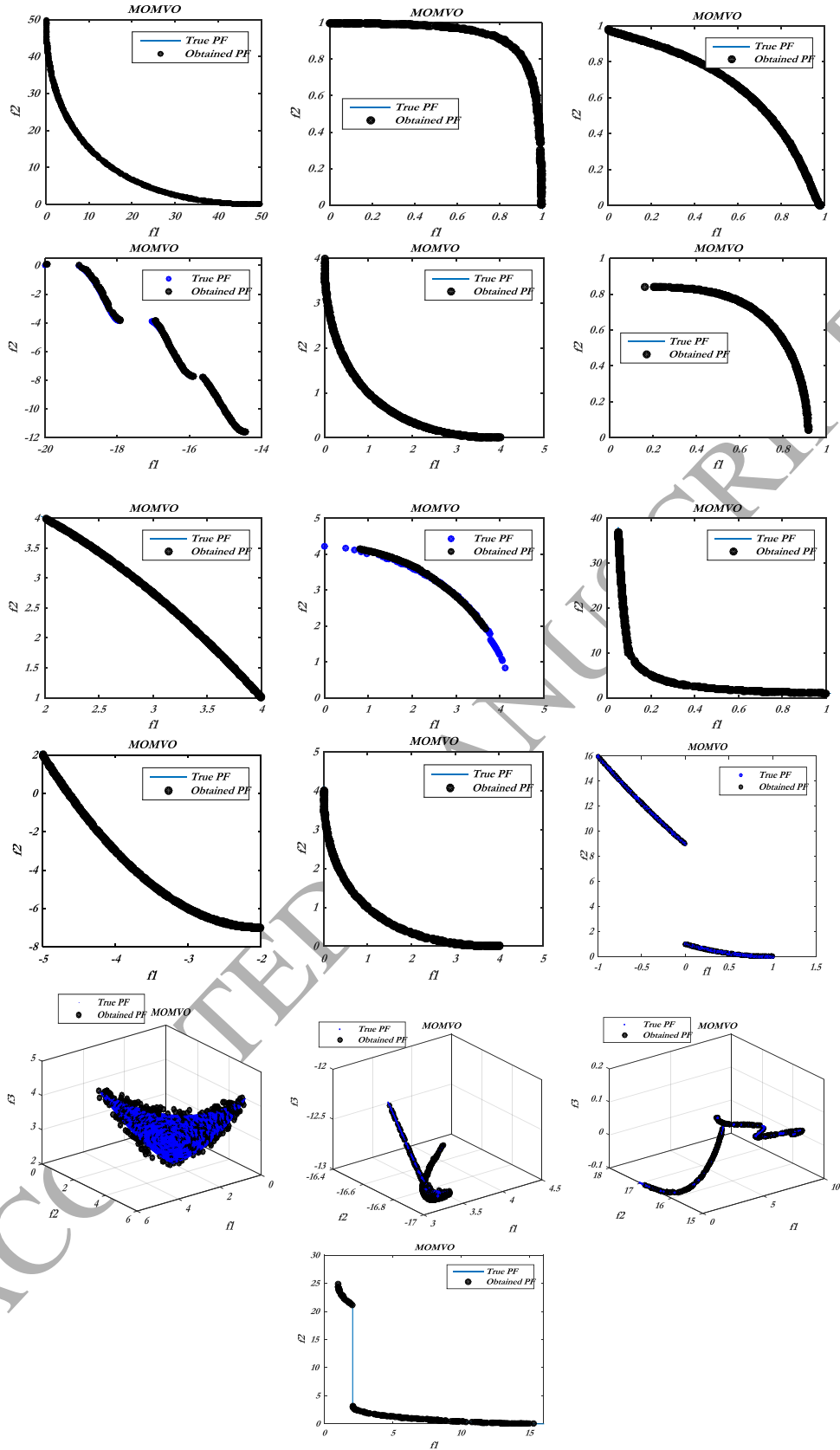


Figure A.6. Best Pareto optimal front obtained by the multi-objective MVO algorithm on BINH1, FON1, FON2, KUR, LAU, LIS, MUR, QUAG, REND1, REND2, SCH1, SCH2, VNT1, VNT2, VNT3 and POL problems.

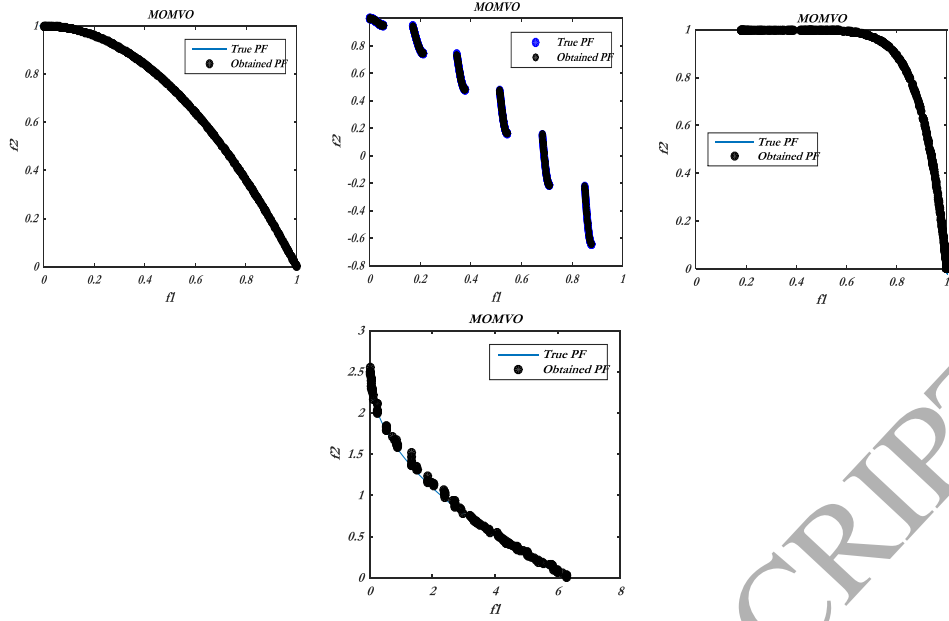


Figure A.7. Best Pareto optimal front obtained by the multi-objective MVO algorithm on DEB1, DEB2, DEB3 and OKA1 problems.

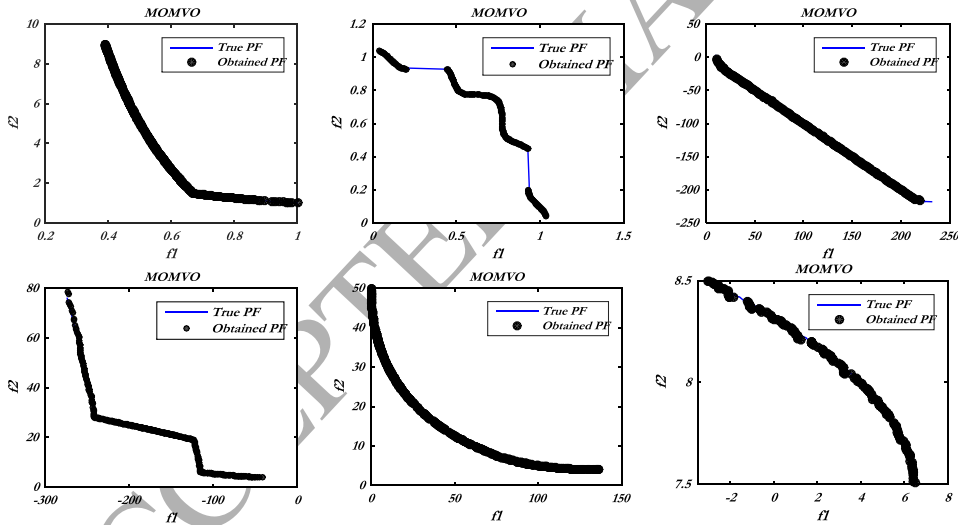


Figure A.8. Best Pareto optimal front obtained by the multi-objective MVO algorithm on CONSTR, TNK, SRN, OSY, BNH and KITA problems.

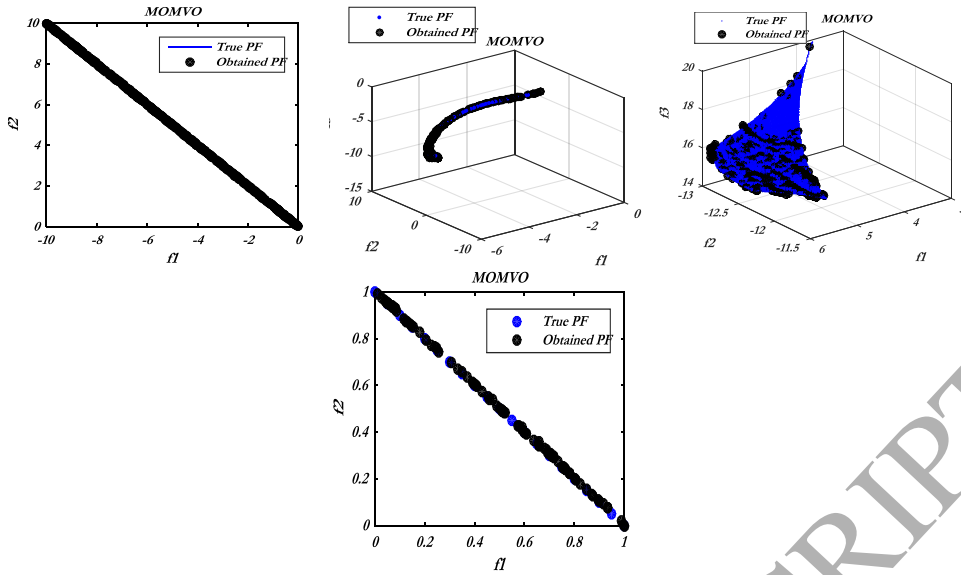


Figure A.9. Best Pareto optimal front obtained by the multi-objective MVO algorithm on BEL, BNH3, VNT4 and CF1 problems.

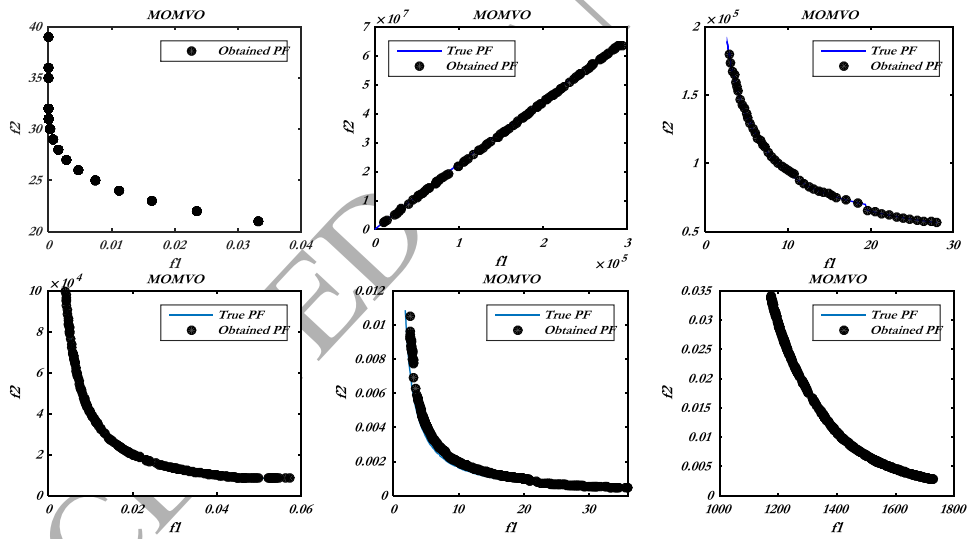


Figure A.10. Best Pareto optimal front obtained by the multi-objective MVO algorithm on Gear Train Design, Pressure Vessel Design, Helical Spring Design, Two Bar Truss Design, Welded Beam Design and Four-bar truss design problems.

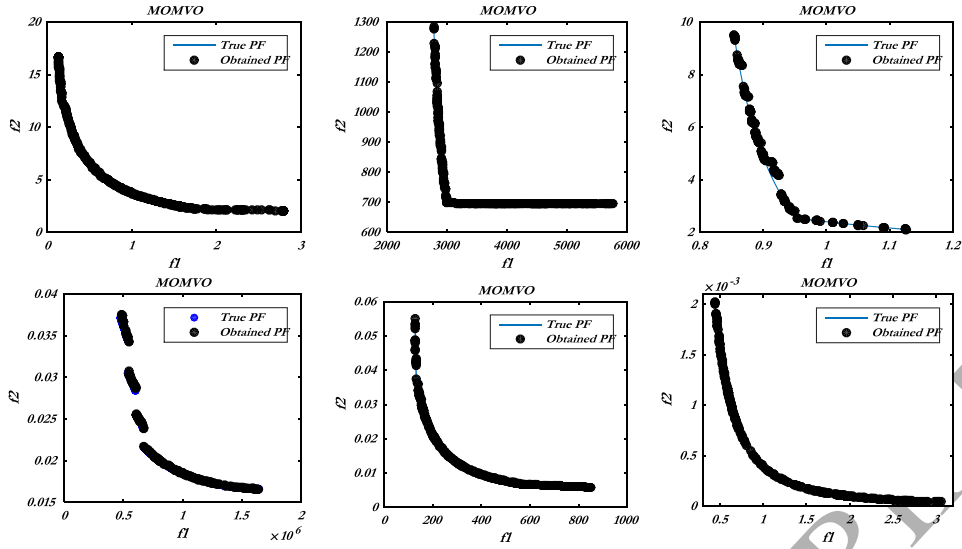


Figure A.11. Best Pareto optimal front obtained by the multi-objective MVO algorithm on Disk Brake Design, Speed reducer Design, CNC Machine Tool Design, Tool Spindle Design, I-Beam Design and Cantilever Beam Design problems.

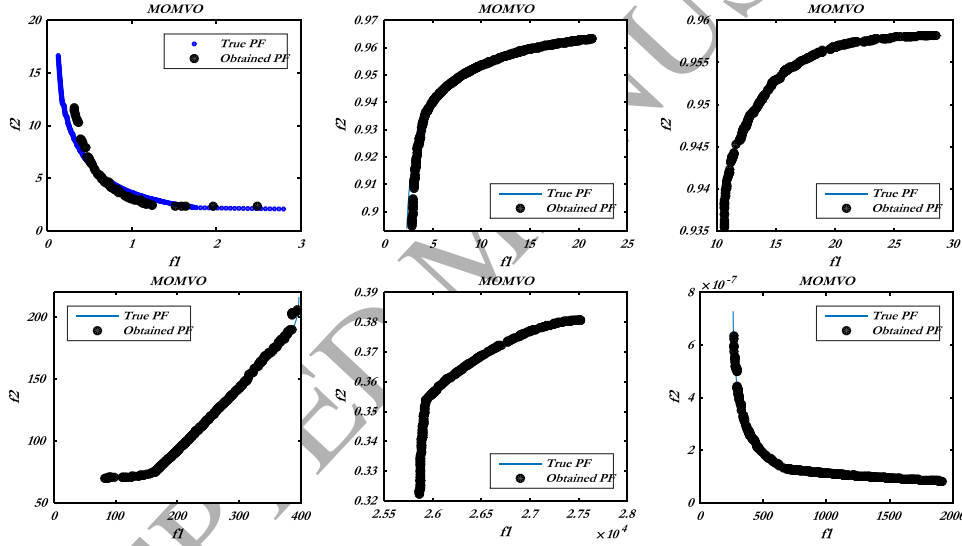


Figure A.12. Best Pareto optimal front obtained by the multi-objective MVO algorithm on Multiple disk clutch brake Design, Isolated Safety Transformer Design, Brushless DC Wheel Motor Design, Vibrating platform design, Satellite Heat Pipe Design and Three-bar truss design problems.

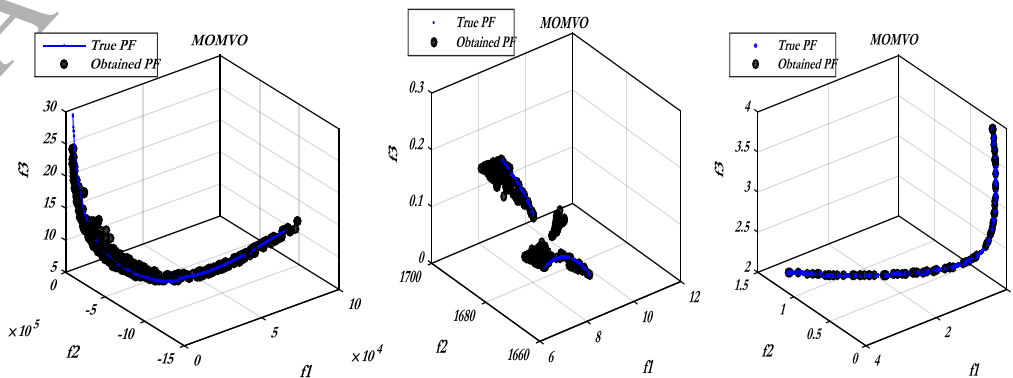


Figure A.13. Best Pareto optimal front obtained by the multi-objective MVO algorithm on Bulk Carrier Design, Car Crash Design and Metal Cutting tool design problems

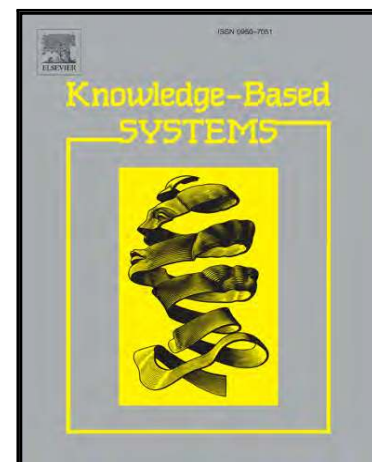
ACCEPTED MANUSCRIPT

Accepted Manuscript

Optimization of Problems with Multiple Objectives using The Multi-Verse Optimization Algorithm

S. Mirjalili , P. Jangir , S.Z. Mirjalili , S. Saremi , I.N. Trivedi

PII: S0950-7051(17)30340-4
DOI: [10.1016/j.knosys.2017.07.018](https://doi.org/10.1016/j.knosys.2017.07.018)
Reference: KNOSYS 3982



To appear in: *Knowledge-Based Systems*

Received date: 19 September 2016
Revised date: 14 July 2017
Accepted date: 15 July 2017

Please cite this article as: S. Mirjalili , P. Jangir , S.Z. Mirjalili , S. Saremi , I.N. Trivedi , Optimization of Problems with Multiple Objectives using The Multi-Verse Optimization Algorithm, *Knowledge-Based Systems* (2017), doi: [10.1016/j.knosys.2017.07.018](https://doi.org/10.1016/j.knosys.2017.07.018)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.